boundaries. The sculpt mode gravity factor is also applied on the cloth physics.

The main brushes and tools for this feature are the Cloth Brush and Cloth Filter, but other transform brushes like Pose and Boundary also support cloth sculpting in the brush settings.

A demo file for trying out the various brushes and tools is available here.

Brushes

Brushes for *Sculpt Mode* bundled in the *Essentials* library.

- Brushes
- Draw
- Draw Sharp
- Clay
- Clay Strips
- Clay Thumb
- Layer
- Inflate
- Blob
- Crease
- Smooth
- Flatten
- Fill
- Scrape
- Scrape Multiplane
- Pinch
- Grab
- Elastic Deform
- Snake Hook
- Thumb
- Pose
- Nudge
- Rotate
- Relax Slide
- Boundary
- Cloth
- Density
- Mask
- Draw Face Sets
- Erase Multires Displacement
- Smear Multires Displacement

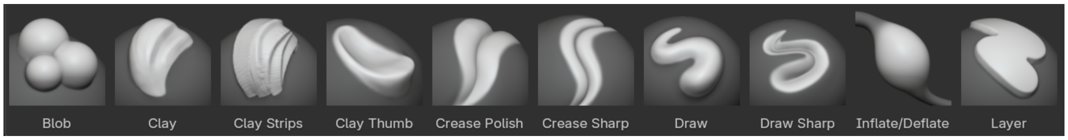
- Paint
- Smear

Brushes

This is a list of all provided ‘Essentials’ brush assets that come with Blender. These are based on various Brush Types which are mentioned for each brush..

Add/Subtract Brushes

These brushes generally push vertices outwards and inwards and are the most customizable to achieve a wide variety of effects. They typically don’y use a color in their thumbnail.


Blob
Clay
Clay Strips
Clay Thumb
Crease Polish
Crease Sharp
Draw
Draw Sharp
Inflate/Deflate
Layer

Draw

Brush Type: Draw
Shortcut: `V`

The standard brush for pushing vertices inwards and outwards from the surface direction.

Draw Sharp

Brush Type: Draw Sharp
Shortcut: `Shift V`

Same as *Draw* but with a much sharper Falloff. Useful for creating creases and sharp angles.

Clay
Brush Type: Clay

Similar to the *Draw* brush but with a flattening effect and subtle smoothing. Useful for polishing and building volumes.

Clay Strips

Brush Type: Clay Strips
Shortcut: `C`

The same as the *Clay* brush, but more aggressive with a square falloff. A common standard for building rough volumes.

Clay Thumb

Brush Type: Clay Thumb

The same as the *Clay* brush, but specifically for emulating the effect of running your thumb over surfaces. Pushes geometry in and sideways.

Layer
Brush Type: Layer

Draw with a fixed height. Useful for adding flat layers to a surface.

Inflate/Deflate

Brush Type: Inflate
Shortcut: `I`

Moves the mesh in multiple direction. Useful for inflating or shrinking surfaces and volumes.

Blob
Brush Type: Blob

Magnifies the mesh as you draw. Useful for an additional inflation effect on the stroke.

Crease Polish

Brush Type: Crease
Shortcut: `Shift C`

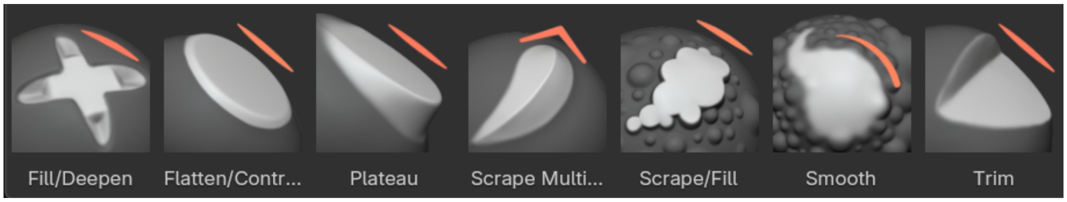
A Draw brush with a pinching effect. Useful for polishing existing creases or carefully creating new ones.

Crease Sharp

Brush Type: Crease

Much sharper and stronger Crease brush. Great for creating thin and deep pinches.

Contrast Brushes



Recognizable by their red thumbnail and cursor. These brushes generally flatten or heighten the contrast of the surface.

Smooth

Brush Type: Smooth
Shortcut: `S`

Smooths out irregularities in the surface and shrinks volumes by averaging the vertices positions. An essential brush that is frequently used.

Flatten/Contrast

Brush Type: Flatten

Pushes vertices to an average height to create a flat surfaces. Alternatively pushes them away from the center for more contrast.

Plateau

Brush Type: Flatten

Similar to Flatten but with a locked orientation and depth to create a consistently flat surface.

Fill/Deepen

Brush Type: Fill

Pushes surfaces upwards towards a flat plane. Useful for filling in holes and crevices. Alternatively deepens existing holes when holding ‘Ctrl’.

Scrape/Fill

Brush Type: Scrape
Shortcut: `Shift T`

Pushes surfaces inwards. Alternatively fills surfaces while holding ‘Ctrl’. This is the most common brush for flattening meshes.

Trim

Brush Type: Scrape

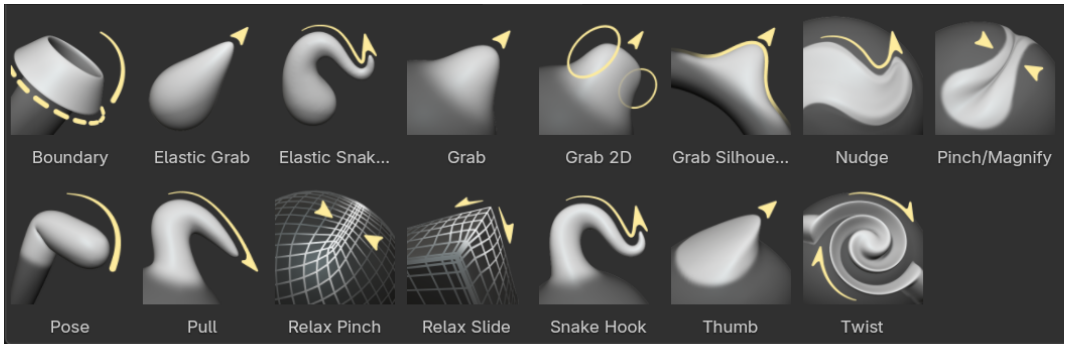
Pushes surfaces inwards toward a locked direction. The depth can be defined by going deeper towards surfaces along the stroke.

Scrape Multiplane

Brush Type: Scrape Multiplane

Scrapes the mesh with two angled planes at the same time, producing a sharp edge between them.

Transform Brushes


Boundary
Elastic Grab
Elastic Snak...
Grab
Grab 2D
Grab Silhoue...
Nudge
Pinch/Magnify
Pose
Pull
Relax Pinch
Relax Slide
Snake Hook
Thumb
Twist

Recognizable by their yellow icon and cursor. These brushes generally move, pinch and magnify the mesh.

Pinch/Magnify

Brush Type: Pinch
Shortcut: `P`

Pulls vertices towards the center of the brush. Useful for polishing angles and creases. Alternatively pushes them away from the center.

Grab

Brush Type: Grab
Shortcut: `G`

Moves vertices along with the mouse. An essential brush for building shapes and adjusting proportions.

Grab 2D

Brush Type: Grab

Similar to Grab but with an infinitely projected falloff. Useful for grabbing broader shapes and giving a similar feel to using Liquify tools in image painting applications.

Grab Silhouette

Brush Type: Grab

Similar to Grab but only affects vertices with the normal facing sideways away from the view. Very useful for adjusting outer silhouettes of thin objects.

Elastic Grab

Brush Type: Elastic Deform

Used to simulate realistic deformations from grabbing of Elastic objects.

Elastic Snake Hook

Brush Type: Snake Hook

Similar to Elastic Grab but rotates affected geometry based on the stroke direction.

Snake Hook

Brush Type: Snake Hook
Shortcut: K

Pulls vertices along with the stroke to create long, snake-like forms. Geometry is rotated and magnified to allow continuous pulling. Much more useful while having Dyntopo enabled.

Pull

Brush Type: Snake Hook

Iteratively picks up and lets go of geometry like the Snake Hook, but much softer. Useful for subtle small scale deforming over longer strokes.

Thumb

Brush Type: Thumb

Same as *Grab* but moves vertices along the surface direction. Useful for preserving specific surfaces.

Pose

Brush Type: Pose

Simulating an armature-like deformations. Useful for quick posing and transformations.

Nudge

Brush Type: Nudge

Similar as *Thumb* but dynamically picks up vertices like the *Snake Hook*. Useful for nudging something along the mesh surface.

Twist

Brush Type: Rotate

Rotates vertices within the brush in the direction mouse.

Relax Slide

Brush Type: Relax Slide

Slides the topology of the mesh in the direction of the stroke while preserving the geometrical shape of the mesh. Alternatively smoothes the mesh on ‘Shift’. Also useful for redistributing topology where it is needed.

Relax Pinch

Brush Type: Relax Slide

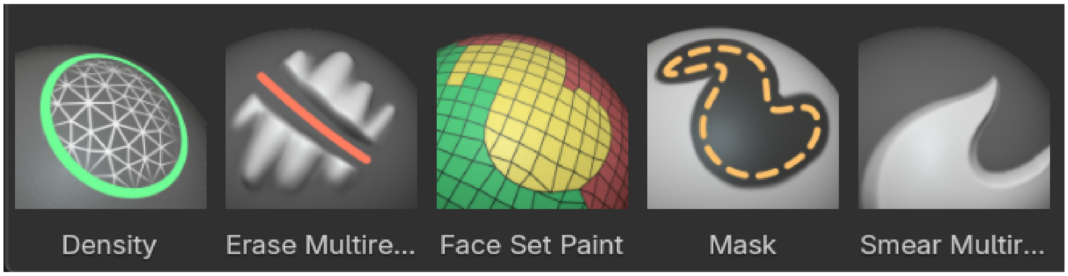
Similar to the Relax Slide brush but pinches/relaxes geometry instead.

Boundary

Brush Type: Boundary

Transform specifically mesh boundaries with various deformations.

Utility Brushes



No clear color assignment. These brushes are general purpose brushes or specific.

Density

Brush Type: Density

Cleans up geometry by collapsing short edges. Specifically for use with Dyntopo.

Mask

Brush Type: Mask
Shortcut: `M`

Paints a selection on parts of the mesh to be unaffected by other brushes.

Draw Face Sets

Brush Type: Draw Face Sets

Paint new, smooth or extend existing Face Sets.

Erase Multires Displacement

Brush Type: Erase Multires Displacement

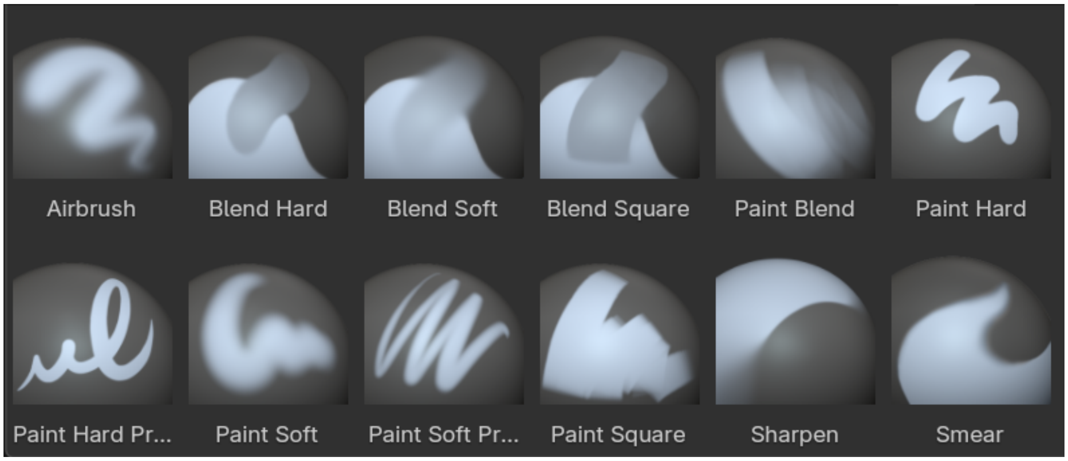
Remove displacement information on a Multiresolution modifier.

Smear Multires Displacement

Brush Type: Smear Multires Displacement

Smear displacement information on a Multiresolution modifier.

Painting Brushes


Airbrush
Blend Hard
Blend Soft
Blend Square
Paint Blend
Paint Hard
Paint Hard Pr...
Paint Soft
Paint Soft Pr...
Paint Square
Sharpen
Smear

Recognizable by their blue thumbnails. These brushes are used for painting color attributes within sculpt mode.

Paint Hard

Brush Type: Paint

A simple hard round falloff.

Paint Soft

Brush Type: Paint

A soft round falloff with pressure sensitivity for only the strength.

Paint Hard Pressure

Brush Type: Paint

A hard round falloff with pressure sensitivity for the brush radius.

Paint Soft Pressure

Brush Type: Paint

A soft round falloff with pressure sensitivity for both radius and strength.

Paint Square

Brush Type: Paint

A hard square brush falloff.

Airbrush

Brush Type: Paint

A soft round brush that builds up over time instead of stroke distance.

Blend Hard

Brush Type: Paint

Similar to Average brushes in other modes with a hard round falloff. Used to blend colors along the stroke.

Blend Soft

Brush Type: Paint

Same as Blend Hard but with a soft round falloff.

Blend Square

Brush Type: Paint

Same as Blend Hard but with a hard square falloff.

Paint Blend

Brush Type: Paint

A mix of a Paint and Blend brush. On low pen pressure the brush averages colors and with high pen pressure it paints colors.

Smear

Brush Type: Smear

Smears colors along the stroke.

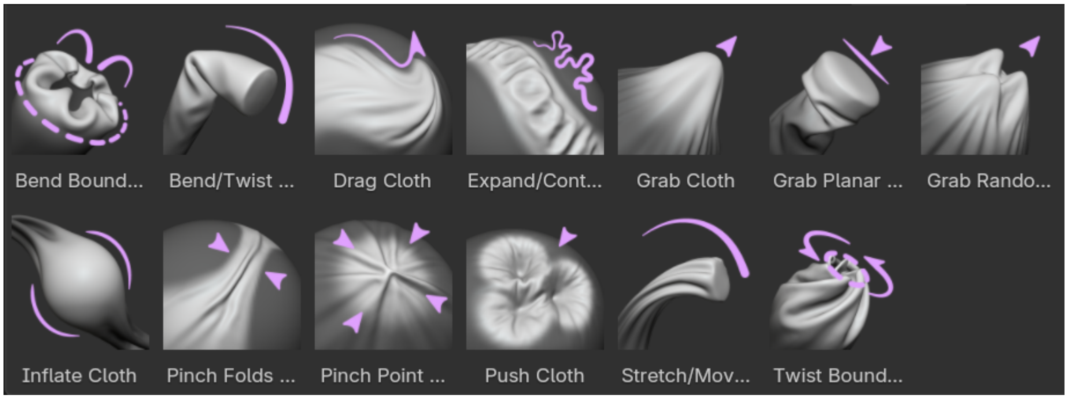
Sharpen

Brush Type: Smear

Pinches the colors inwards to create sharp edges or points.

Simulation Brushes

These brushes are similar to regular brushes but with an additional cloth simulation applied. These are ideally used on a relatively low resolution, since the mesh density defines the size of cloth dynamics.


Bend Bound...
Bend/Twist ...
Drag Cloth
Expand/Cont...
Grab Cloth
Grab Planar ...
Grab Rando...
Inflate Cloth
Pinch Folds ...
Pinch Point ...
Push Cloth
Stretch/Mov...
Twist Bound...

Drag Cloth

Brush Type: Cloth

Nudges the geometry along the surface while minimally affecting the overall shape of the object.

Push Cloth

Brush Type: Cloth

Pushes geometry inwards or outwards.

Grab Cloth

Brush Type: Cloth

Grabs geometry within the brush radius firmly, while surrounding geometry is being simulated to follow.

Grab Planar Cloth

Brush Type: Cloth

Similar to Grab Cloth but with a line as the brush radius instead of a circle.

Grab Random Cloth

Brush Type: Cloth

Similar to Grab Cloth but with a noise texture applied to create more random variation.

Inflate Cloth

Brush Type: Cloth

Inflates the geometry outwards or inwards.

Expand/Contract Cloth

Brush Type: Cloth

Creates compression or stretching on geometry.

Pinch Point Cloth

Brush Type: Cloth

Pinches geometry to the center point of the radius, creating folds from all sides.

Pinch Folds Cloth

Brush Type: Cloth

Pinches only from two perpendicular sides along the stroke direction, creating parallel folds along the stroke.

Bend/Twist Cloth

Brush Type: Pose

A pose brush that rotates geometry.

Stretch/Move Cloth

Brush Type: Pose

A pose brush that translates and scales geometry.

Bend Boundary Cloth

Brush Type: Boundary

Bend only open boundaries of the mesh, folding the surrounding geometry in the process.

Twist Boundary Cloth

Brush Type: Boundary

Twist open boundaries of the mesh, creating twisting folds.

Draw

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Moves vertices inward or outward, based the average vertex normals within the brush radius. This is a very default behavior for sculpting and can be used in most cases.

It is common to use this particular brush with heavy customization for creating many custom brushes.

Brush Settings

Note

More info at General brush settings and on Advanced brush settings.

VDM Displacement

Vector Displacement Maps are supported for the *Draw* brush to insert complex & overhanging shapes. Unlike regular displacement, this uses all 3 color channels of the image to displace geometry in three directions instead of just one.

An example of various VDM brushes used on a smooth head from the official demo file.

Download the demo file for more information and to try the feature out.

To use this feature, enable Vector Displacement in the texture panel. All stroke methods are supported, but the recommended behavior is *Anchored*.

Ideal images for vector displacement are open EXR files with color clamping disabled.

Note

This feature is only supported with Area Plane mapping.

Draw Sharp

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Similar to the Draw brush, but it always deforms the mesh from the original coordinates and uses the *Sharper* Falloff by default.

Draw Sharp is useful on high density meshes for creating cloth wrinkles, stylized hair or hard surface edges. To further sharpen or polish sharp edges in the case that the mesh density is not enough, it's recommended to use the Pinch, Crease or Multiplane Scrape brushes.

A limitation is that the brush does not remesh the sculpted surfaces with Dyntopo enabled. Because of that, a better brush to use with Dyntopo can be Crease.

Brush Settings

General

Direction
    On *Subtract* by default to carve in creases. More info at Direction

Note

More info at General brush settings and on Advanced brush settings.

Clay

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Draw brush, but includes settings to adjust the sculpt plane on which the brush acts. That's because it behaves like a combination of the Flatten and *Draw* brushes.

This brush is useful for building and removing volumes and shapes like real clay, because it flattens details as you add/subtract from the surfaces.

If used together with Dyntopo it's easy to continuously build shapes, even in a single stroke.

Brush Settings

General

Hardness
: Slightly higher by default. This makes the profile of the brush more noticeable. More info at Hardness

Auto-Smooth
: Enabled by default for a consistent smoothing effect. With lower brush strength (for example with lower pen pressure) the smoothing effect will be more noticeable and can be used to create and then blend/polish shapes in a single stroke. Enable Pressure to modulate the use of auto-smooth even more with pen inputs. More info at Auto-Smooth

Note

More info at General brush settings and on Advanced brush settings.

Clay Strips

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Similar to the Clay brush, but it uses a square tip shape instead of a round one.

Just like the *Clay* brush, it's useful for building and removing volumes and shapes like real clay, because it flattens details as you add/subtract from the surfaces.

Clay Strips is very commonly used for aggressive building of volumes and deliberate control over shapes on the surface. This brush alone can be used for a fast rough pass over the entire sculpt, with additional smoothing or polishing often required afterwards. This brush can be very versatile with varying stroke directions, repeated strokes and pen pressure to achieve various results.

If used together with Dyntopo it's easy to continuously build shapes, even in a single stroke.

Brush Settings

General

Normal Radius
: Higher by default. This ensures that the brush does not change directions to sporadically during a stroke. More info at Normal Radius

Tip Roundness

Very low by default for a square shape for more deliberate shaping. More info at Tip Roundness

Note

More info at General brush settings and on Advanced brush settings.

Clay Thumb

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Clay brush. It imitates the effect of deforming clay with the finger, accumulating material during the stroke. The sculpt plane tilts during the stroke in the front part of the brush to achieve this effect.

Brush Settings

Note

More info at General brush settings and on Advanced brush settings.

Layer

Reference

Mode:: Sculpt Mode
Brush:: *Sidebar ▸ Tool ▸ Brush Settings ▸ Advanced ▸ Brush Type*

This brush is similar to Draw, except that the height capped. This creates the appearance of a flat layer.

It is recommended to use the Persistent setting and regularly Set Persistent Base, so that multiple strokes to not add on top of each other.

Brush Settings

General

Hardness
    Higher by default to ensure the profile of layers is more noticeable. More info at Hardness

Note

More info at General brush settings and on Advanced brush settings.

Unique

Height
    The fixed height of each stroke. This is measured using the scene scale, so it is consistent no matter the amount of zoom or or object size.

Persistent

This will ensure that multiple strokes use the same height, as if sculpting a single layer.

Set Persistent Base

This button resets a new base so that you can sculpt new layer.

Inflate

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to Draw, except that vertices are moved in the direction of their own normals. Especially useful when sculpting meshes with a lot of curvature.

Also available as a Mesh Filter to inflate all unmasked areas at once.

Brush Settings

General

Direction
  Either Inflate or Deflate sculpted areas. This is different from the typical Add & Subtract.

Note

More info at General brush settings and on Advanced brush settings.

Blob

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to Draw, but vertices are pushed outwards like an inverted pinching effect. This will lead to a more consistent spherical curvature and thickening of strokes.

Brush Settings

General

Magnify
: By default at 0.5 to push out the mesh during the stoke. More info at Pinch/Magnify

Note

More info at General brush settings and on Advanced brush settings.

Crease

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Create sharp indents or ridges by pushing or pulling the mesh, while pinching the vertices together.

Crease can also be used to sharpen and polish existing creases. Enable pressure sensitivity on Strength to regulate the add/subtract effect while pinching creases.

Brush Settings

General

Pinch

Adds a consistent pinching effect to your stroke. If set to 0 it the brush will behave like the Draw brush. If set to 1 and the brush strength set to 0, the brush will behave like a Pinch brush.

Note

More info at General brush settings and on Advanced brush settings.

Smooth

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Smooths the positions of the vertices to either polish surfaces or remove volume from larger shapes. Because this brush is so essential, it's always accessible by holding `Shift` and sculpting.

Also available as a Mesh Filter to smooth all unmasked areas at once.

Note

The brush called *Smooth* will be used whenever holding `Shift` and sculpting. If the smoothing strength or behavior needs to be changed, switch to the *Smooth* brush and adjust the settings there.

Brush Settings

General

Strength

The strength of the smoothing is relative to the density of the mesh. If the resolution is increased on the sculpted mesh, the strength of the smooth brush will be weaker than before and needs to be increased.

Direction `Ctrl`

Smooth

Smooths the surface of the mesh.

Enhance Details

Enhances details on the surface of the mesh by applying a smoothing operation in the opposite direction.

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation

There are two deformation types.

Laplacian::	Smooths the surface and volumes. This is the default behavior.
Surface::	Smooths only the surface of the mesh, while preserving the volume.

Shape Preservation

How much of the original shape is preserved while smoothing. Increasing the value reduces the effect of having multiple iterations on the strength of smoothing.

Per-Vertex Displacement

How much the position of each individual vertex influences the final result. Increasing the value reduces the overall strength of smoothing.

Iterations

Number of smoothing iterations per brush step.

Note

This method works by applying regular smoothing, computing the difference between the original (blended between start of iteration and fully original based on *Shape Preservation*) and the smoothed mesh, smoothing these offsets, pushing vertices back using the smoothed offsets, and finally blending in the original mesh based on *Per-Vertex Displacement*.

Flatten

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Flatten or contrast surfaces by pulling and pushing them towards (or away from) a defined medium height. This medium height is determined via an area plane within the brush radius.

This area plane height can be further defined in the brush settings.

Brush Settings

General

Direction `Ctrl`

Invert the direction to push surfaces away from the sculpt plane, creating more surface contrast as a result.

Note

More info at General brush settings and on Advanced brush settings.

Fill

Reference

Mode:: Sculpt Mode
Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Flatten brush, but only pushes surfaces upwards to the medium height.

Although `Ctrl` can be held to invert the effect to a Scrape brush, if *Invert to Scrape* is enabled. When disabled, the inverted direction will push surfaces away.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Invert to Scrape
When enabled, holding `Ctrl` while sculpting changes the brush behavior to be the same as the *Scrape* brush. When disabled, holding `Ctrl` while sculpting, will push vertices below the cursor downward.

Scrape

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Flatten brush, but only pushes surfaces downward to the medium height.

Although `Ctrl` can be held to invert the effect to a Fill brush, if *Invert to Fill* is enabled. When disabled, the inverted direction will push surfaces away.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Invert to Fill

When enabled, holding `Ctrl` while sculpting changes the brush behavior to be the same as the *Fill* brush. When disabled, holding `Ctrl` while sculpting, will push vertices above the cursor up away from the cursor.

Scrape Multiplane

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Scrapes the mesh with two angled planes at the same time, creating a sharp edge between them. This is useful for creating & polishing hard surface objects.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Plane Angle
: The angle between the two planes of the brush, pressing `Ctrl` inverts the angle.

Dynamic Mode
: When enabled, the angle is dynamically updated based on the surface under the brush. The *Plane Angle* then controls how much the angle will increase when applying pen pressure. When pressing `Ctrl`, it locks the plane angle to 0 degrees.

Show Cursor Preview

Displays a preview of the two planes and the angle they form during the stroke.

Pinch

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ▸ *Tool* ▸ *Brush Settings* ▸ *Advanced* ▸ *Brush Type*

Pulls geometry towards the center of the brush. When inverted via `Ctrl` it will *Magnify* geometry by pushing them away from the center of the brush.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

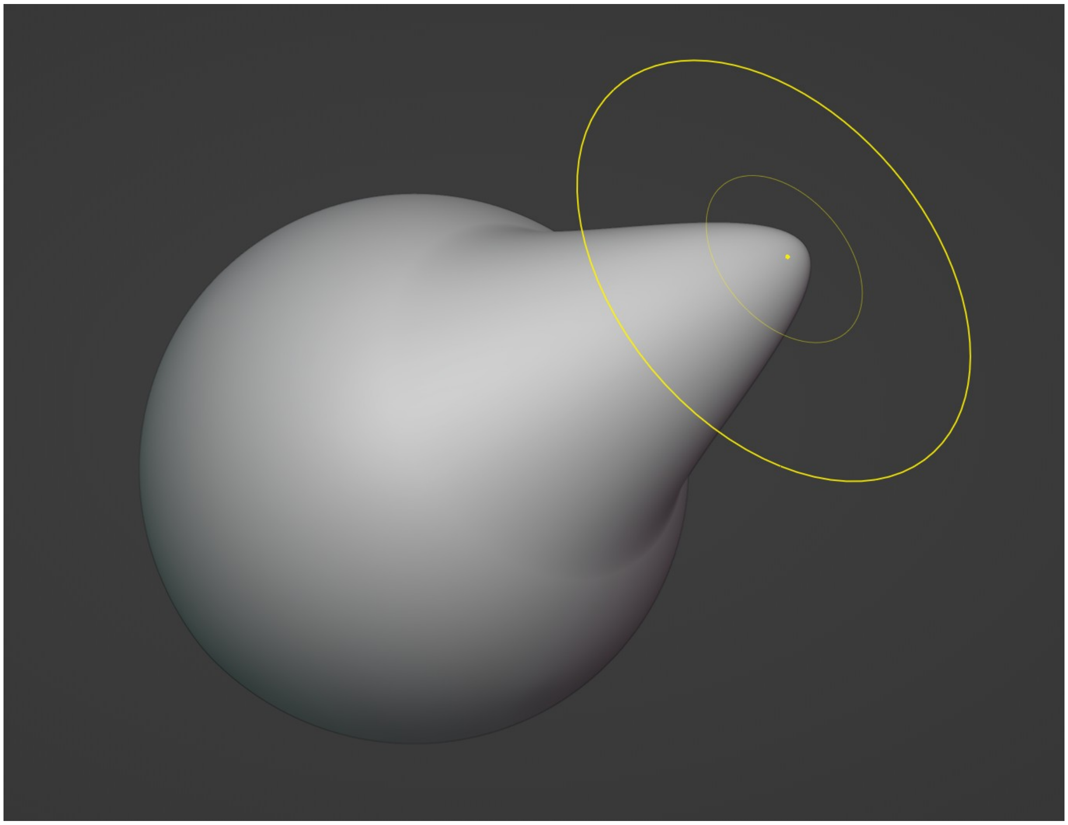
Grab

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Drag geometry across the screen, following the cursor. *Grab* only moves the vertices that are under the brush radius at the start of the stroke. This is an essential sculpting brush to be used frequently to build shapes and adjust proportions.



Note

The effect is similar to moving geometry in Edit Mode with Proportional Editing enabled, except that *Grab* can make use of other Sculpt Mode options and brush settings.

Brush Settings

General

Radius
: Pressure Sensitivity is not supported for this brush type. More info at Radius.

Strength
: Pressure Sensitivity is not supported for this brush type. More info at Strength.

Normal Radius
: For this brush, this setting is a purely visual change. It does not alter the brush behavior. More info at Normal Radius.

Auto-Smooth
: This setting is not supported. More info at Auto-Smooth.

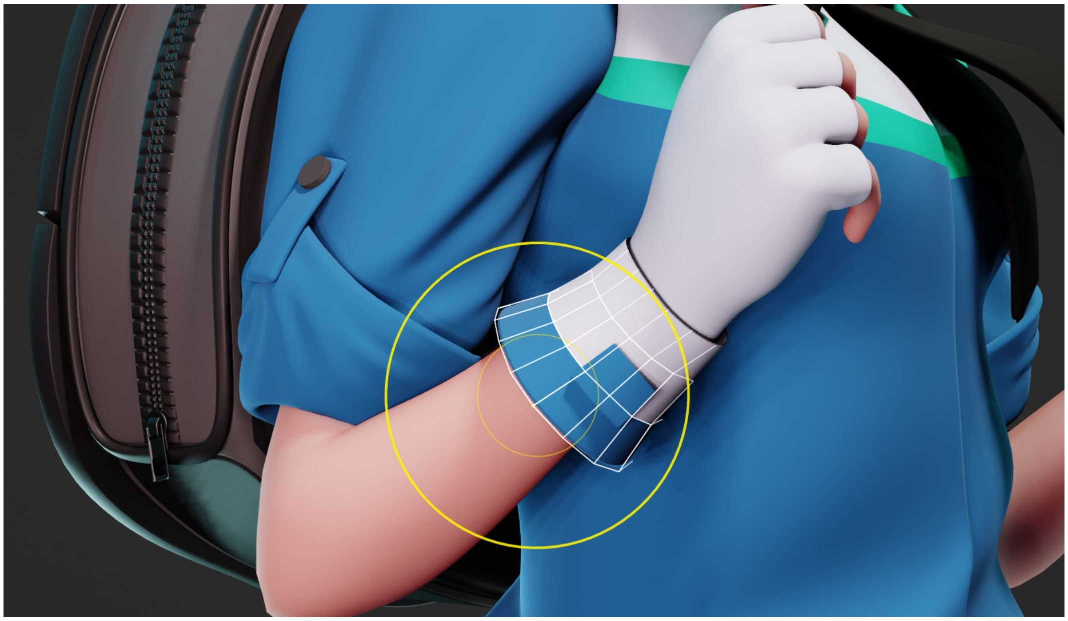
Note

More info at General brush settings and on Advanced brush settings.

Unique

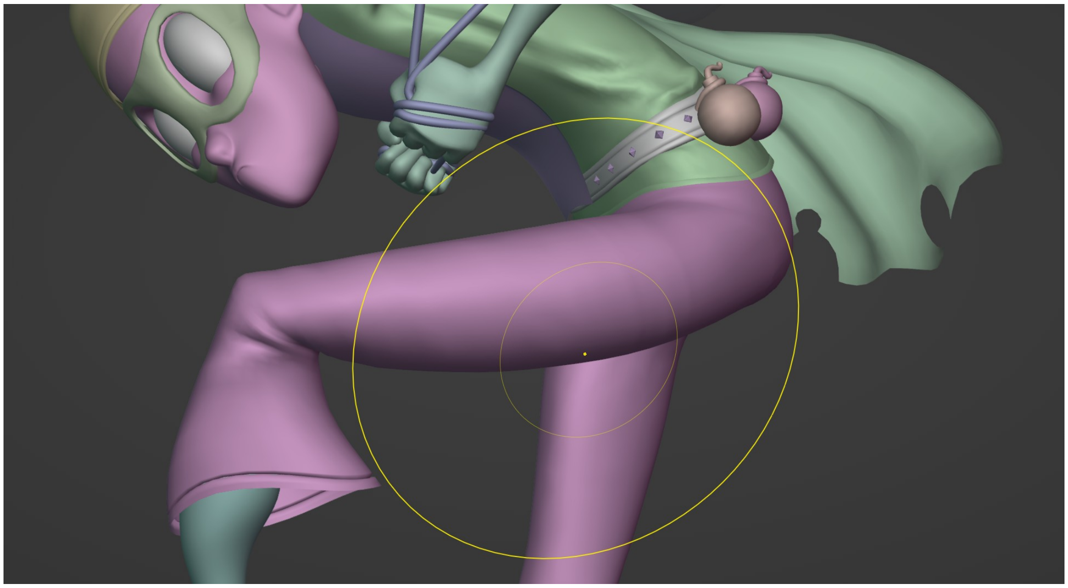
Grab Active Vertex
: Applies the maximum strength of the brush to the highlighted active vertex, making it easier to manipulate low-poly models or meshes with modifiers.

Enabling this option also enables a white wireframe overlay within the brush radius. This helps to visualize the real base geometry that is being manipulated while sculpting with Modifiers.



Grab Silhouette

Preserves the object’s silhouette shape by only grabbing vertices on one side of the mesh curvature. The shape of the silhouette is determined by the orientation of the 3D Viewport and the start of the stroke.



Note how in the image only the bottom side of the leg is pulled down, despite the size of the brush.

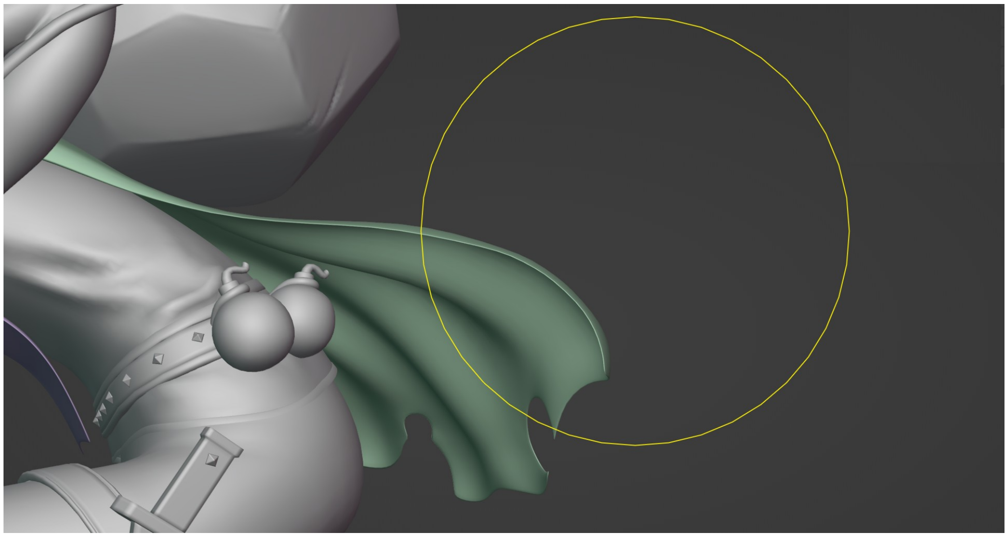
Tip

This setting is also useful for grabbing a single side of a crease and pushing it further inwards, creating a more pinched crease.

Additional Workflows

2D Grab Brush

If the Falloff Shape is set to *Projected*, the brush can grab infinitely deep into the viewport. This is especially useful for much broader changes to a sculpt.



The stroke can also be started outside of the mesh (like in empty 3D space) and grab the vertices within the brush radius. This can be useful for sculpting flat and tube-like meshes.

Elastic Deform

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ▸ *Tool* ▸ *Brush Settings* ▸ *Advanced* ▸ *Brush Type*

Used to simulate realistic deformations such as grabbing or twisting of Elastic objects. For example, this tool works great for modeling the shape of organic objects such as humans or animals. When pressing `Ctrl`, the brush deforms vertices along the normal of the active vertex.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation

The surface alteration that is used in the brush.

Grab::	Used to drag a group of vertices around.
Bi-scale Grab::	Like *Grab* but the falloff is more localized to the center of the brush.
Tri-scale Grab::	Like *Bi-scale Grab* but the falloff is more localized to the center of the brush.
Scale::	Displaces vertices away from the active vertex.

Twist:: Vertices are rotated around the active vertex.

Volume Preservation

Higher values preserve volumes more, but also lead to more bulging. (This value determines the poisson ratio for elastic deformation)

Snake Hook

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Pulls vertices along with the movement of the brush to create long, snake-like forms. During the stroke, geometry will be dynamically picked up & let go.

When the *Rake* setting is used, the brush can also be used to rotate geometry via dragging.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Magnify
: Pulled geometry tends to lose volume along the stroke. With *Magnify* value greater than 0.5 this is prevented. More info at Pinch/Magnify

Rake
: Rotates geometry along the direction of the stroke.

Deformation

Deformation type that is used by the brush.

Radius Falloff::	Applies the brush falloff to the tip of the brush.
Elastic::	Modifies the entire mesh using an Elastic deformation. More info in the Elastic Deform brush.

Thumb

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Grab brush, but instead only moves the geometry along the surface (using the area plane). This is useful for grabbing surfaces with a very specific direction, or without making too impactful changes to the overall object shape.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Pose

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ‣ *Tool* ‣ *Brush Settings* ‣ *Advanced* ‣ *Brush Type*

Deform a model simulating armature-like workflow. This can either be useful for posing a model without a rig, adjusting the proportions of a mesh or other fast deformations.

The brush will automatically determine an origin point, indicated with a while line on the brush cursor.

If the Deformation Target is changed, the brush can also be used for cloth sculpting.

Brush Settings

General

Only Radius and Auto-Masking has an impact on the brush behavior for this brush.

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation

Deformation type that is used by the brush.

Rotate/Twist::	Rotates the mesh around the pose origin. When pressing `Ctrl`, the brush applies a twist rotation instead (and disables any IK segments that are used).
Scale/Translate::	Scale the mesh based on the pose origin. While holding `Ctrl` the brush moves the mesh.
Squash/Stretch::	Works similar to *Scale/Translate* however, it applies different scale values along different axes to achieve the stretching effect.

Rotation Origins

Method to set the rotation origin for the pose origin or individual IK segments.

Topology::	Sets the rotation origin automatically using the topology and shape of the mesh.
Face Sets::	Creates a pose segment per Face Set, starting from the active face set. This can lead to the most accurate and desirable results.
Face Sets FK::	Simulates a Forward Kinematics deformation using the Face Set under the cursor as control.

Pose Origin Offset

Offset of the pose origin in relation to the brush radius. This is useful to manipulate areas with a lot of complex shapes like fingers.

Smooth Iterations

Controls the smoothness of the falloff of the deformation.

Pose IK Segments

Controls how many IK segments are going to be created for posing. This can be seen by a divided white line on the cursor. This is also useful for making curved deformations with the pose brush, like hair clumps and tails.

Lock Rotation when Scaling

When using *Scale/Translate Deformation*, do not rotate the segment; only scaling is applied.

Keep Anchor Point

Keeps the position of the last segment in the IK chain fixed. If this is disabled, the mesh can be dragged around more freely, creating snake like shapes.

Connected Only

The brush will only affect topologically connected elements. Disabling this will allow deforming multiple disconnected meshes at the same time, for example characters with clothing & shoes.

Disabling this setting can have a big impact on performance, as neighboring elements will be merged internally. Keeping the *Max Element Distance* as low as possible will help counteract the performance impact.

Max Element Distance

Maximum distance to search for disconnected loose parts in the mesh.

Nudge

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Moves vertices in the direction of the brush stroke.

Similar to the Snake Hook brush, but instead only moves the geometry along the surface (using the area plane). This is useful for grabbing geometry along curved surfaces, without making too impactful changes to the overall object shape.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Rotate

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Rotates geometry in the direction in which the cursor is moved. The initial drag direction is the zero angle and by rotating around the center you can create a vortex/swirl effect.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Relax Slide

Reference

Mode:: Sculpt Mode

Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

This brush deforms the topology of the mesh while minimizing changes to the geometrical shape of the mesh. By default it will drag geometry, but this can be changed in the *Deformation* settings.

This brush is especially useful for redistributing topology to areas that require more detail, or sliding geometry to somewhere where they should be.

Holding `Shift` changes the brush effect to Relax geometry, creating an even distribution of topology.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation
    Deformation type that is used by the brush.

Drag::	Slides the topology of the mesh in the direction of the stroke.
Pinch::	Slides the topology of the mesh towards the center of the stroke.
Expand::	Slides the topology of the mesh away from the center of the stroke.

Boundary

Reference

Mode:: Sculpt Mode
Brush:: *Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Similar to the Pose brush but deforms the open boundaries of a mesh. The tool detects the mesh boundary closest to the active vertex and propagates the deformation using the brush Falloff into the mesh.

The main use cases of this brush are the *Bend* and *Expand* geometry, which leads to the best results on evenly distributed quad based topology. Use the *Inflate*, *Grab*, *Twist*, and *Smooth* deformation modes, to further adjustments and tweaks to the result (which do not depend that much on a clean topology).

Tip

Boundaries to hidden geometry will also be counted as an open boundary.

The boundary origin is displayed via a white line, which indicates the reach of the deformation. The targeted boundary that will be deformed is highlighted in the brush cursor color.

If the Deformation Target is changed, the brush can also be used for cloth sculpting.

Note

Evenly distributed and quad based topology will lead to much better results. Triangles and N-gons are also supported but may lead to unpredictable outcomes.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation

Deformation type that is used by the brush.

Bend::	Rotates the boundary around the local Y axis. Useful for creating folding shapes, like sleeves.
Expand::	Moves/extends the mesh boundary in the local X direction. Useful for extending the boundaries along the surface.
Inflate::	Works similar to the Inflate tool but, the vertices that are inflated are constrained to the mesh boundary.
Grab::	Works similar to the Grab tool but, the vertices that are grabbed are constrained to the mesh boundary.
Twist::	Rotates the active boundary around the local Z axis. Useful for creating folds like on a skirt.
Smooth::	Works similar to the Grab tool but, the vertices that are smoothed are constrained to the mesh boundary.

Boundary Falloff

How the brush Falloff is applied across the boundary.

Constant::	Applies the same deformation in the entire boundary.
Brush Radius::	Applies the deformation only within the brush radius.
Loop::	Applies the brush falloff in a loop pattern along the boundary.
Loop and Invert::	Applies the falloff radius in a loop pattern, inverting the direction back & forth.

Boundary Origin Offset

Offset of the boundary origin in relation to the brush radius.

Cloth

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

This brush simulates cloth physics on the mesh under the brush cursor. There are various deformation types and settings to customize the brush.

It's also easy to sculpt the mesh with other brushes and tools in between using the cloth brushes.

Note

Using a relatively small brush size makes the calculations much faster, while larger brush sizes might be too slow to get a usable brush.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Persistent

Allows the cloth brush to not accumulate deformation after each stroke. This is convenient to always simulate the cloth based on the same initial shape, but applying different forces to it.

When disabled, deformations accumulate after each stroke.

Set Persistent Base

Resets the base mesh so that you can add another layer of deformations.

Simulation Area

Selects the part of the mesh that is going to be simulated when the stroke is active. This can greatly affect performance depending on the complexity of the mesh.

Local

Simulates only a specific area around the brush limited by a fixed radius.

Global

Simulates the entire mesh.

Dynamic

The active simulation area moves with the brush while still being limited by a fixed radius.

Simulation Limit

The Factor added relative to the size of the radius to limit the cloth simulation effects.

Simulation Falloff

The area to apply deformation falloff to the effects of the simulation. This setting is a factor of the *Simulation Limit* and is shown as a dashed line around the cursor.

Pin Simulation Boundary

Lock the position of the vertices in the simulation falloff area to avoid artifacts and create a softer transition with unaffected areas.

Deformation

The type of cloth deformation that is used by the brush.

Drag::	Simulates pulling the cloth to the cursor, similar to placing a finger on a table cloth and pulling.
Push::	Simulates pushing the cloth away from the cursor, similar to placing a finger on a table cloth and pushing.
Pinch Point::	Simulates pulling the cloth into a point.
Pinch Perpendicular::	Simulates pulling the brush into a line.
Inflate::	Simulates air being blown under the cloth so that the cloth lifts up.
Grab::	Simulates picking up and moving the cloth.
Expand::	Simulates stretching the cloth out.
Snake Hook::	Simulates moving the cloth without producing any artifacts in the surface and creates more natural looking folds than any of the other deformation modes. This is accomplished by adjusting the strength of the deformation constraints per brush step to avoid affecting the results of the simulation as much as possible.

Force Falloff

Shape used in the brush to apply force to the cloth.

Radial::	Applies the force as a sphere.
Plane::	Applies the force as a plane.

Cloth Mass

Mass of each simulation particle.

Cloth Damping

How much the applied forces are propagated through the cloth.

Soft Body Plasticity

The amount the cloth preserves its original shape, acting as a Soft Body.

Use Collisions

Enables the detection of collisions with other objects during the simulation. In order for the sculpt object to collide with objects, the collision object must have Collision Physics activated.

Density

Reference

Mode:: Sculpt Mode
Brush:: *Sidebar* ▸ *Tool* ▸ *Brush Settings* ▸ *Advanced* ▸ *Brush Type*

The Density brush is specifically meant for use with Dyntopo to remove/add detail in the mesh.

Even when the Refine Method is set to *Subdivide Edges*, this brush is always able to collapse edges. This ensures that while focusing on adding detail to your sculpt, the *Density* brush can always be used to simplify and polish surfaces.

This brush has no effect if Dyntopo is disabled.

Tip

In combination with auto-smooth the brush can polish surfaces while it remeshes them. On tube-like geometry it can also shrink and dissolve volumes completely.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Mask

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Paint a selection on parts of the mesh to be unaffected by other brushes & tools. The mask values are shown as a gray-scale overlay.

Note

More information in the Masking Introduction.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Mask Tool

The mask brush has two modes:

Draw::	Mask drawing.
Smooth::	Holding `Shift` will instead smooth existing masks.

Draw Face Sets

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

Draw new or extend existing Face Sets with each stroke.

Holding `Ctrl` will continue drawing the same face set as the one under the cursor. Holding `Shift` will relax or smooth the edges of the face sets by modifying the underlying topology so edges flow along the perimeter of the face sets. This will remove the jagged lines visible after drawing or creating a face set.

Note

More information in the Face Set Introduction.

Brush Settings

General

While a lot of the general brush settings are supported, it's not needed to change them from the default, as the brush purpose is very simple.

Note

More info at General brush settings and on Advanced brush settings.

Erase Multires Displacement

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ▸ *Tool* ▸ *Brush Settings* ▸ *Advanced* ▸ *Brush Type*

This brush deletes displacement information of the Multires Modifier, resetting the mesh to a regular subdivision surface result.

This can be used to reset parts of the sculpt or to fix reprojection artifacts after applying a Shrinkwrap Modifier.

Tip

This brush works best after using Apply Base.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Smear Multires Displacement

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar ‣ Tool ‣ Brush Settings ‣ Advanced ‣ Brush Type*

This tool deforms displacement information of the Multires Modifier, moving the displaced vertices without affecting the base mesh.

Smearing effect can be used multiple times over the same area without generating any artifacts in the topology.

Tip

This brush works best after using Apply Base.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation
  Deformation type that is used by the brush.

Drag::	Pulls the displacement values in the direction of the brush.
Pinch::	Pulls the displacement values towards the center of the brush, creating hard surface effects without pinching the topology.
Expand::	Pushes the displacement values away from the brush center, smoothing the displacement.

Paint

Reference

Mode::	Sculpt Mode
Brush::	*Sidebar* ▸ *Tool* ▸ *Brush Settings* ▸ *Advanced* ▸ *Brush Type*

Paints on the active color attribute. Hold `Shift` to blur painted colors instead.

Color attribute's can be managed in the pallette pop-over in the middle of the header.

Note

More information in the Painting Introduction.

Brush Settings

General

Strength

This settings has a different effect on this brush. Instead of defining the strength of each individual step in the stroke, it determines the overall Opacity of the applied color.

Use the *Flow* setting instead for faster increasing of strength.

Note

More info at General brush settings and on Advanced brush settings.

Unique

Flow
Amount of paint that is applied per stroke sample. Used to create fast/slow accumulation effect.

Wet Mix
Amount of paint that is picked from the surface into the brush color. Can achieve the effect of a wet canvas.

Wet Persistence
Amount of wet paint that stays in the brush after applying paint to the surface.

Wet Paint Radius
Ratio between the brush radius and the radius that is going to be used to sample the color to blend in wet paint.

Density
Amount of random elements that are going to be affected by this brush. Use this for a more detailed airbrush effect. This works best on a high resolution.

Tip Scale X
Scale of the brush tip in the X axis. This is useful for a achieving a painting stroke like a marker or paint roller.

Smear

Reference

Mode:: Sculpt Mode
Brush:: *Sidebar ▸ Tool ▸ Brush Settings ▸ Advanced ▸ Brush Type*

Smears the painted colors of the active color attribute. It takes the colors under the cursor, and blends them in the direction of your stroke.

Brush Settings

General

Note

More info at General brush settings and on Advanced brush settings.

Unique

Deformation
- **Drag::** Smear colors along the direction of the stroke.
- **Pinch::** Smear colors inwards towards your brush center.
- **Expand::** Smear colors outwards away from your brush center.

Toolbar

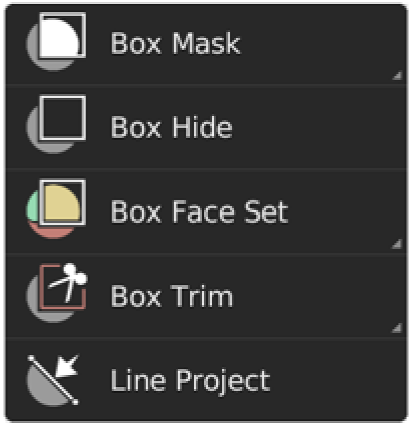
The amount of tools in sculpt mode is very extensive. This is an overview of all of them, categorized by their general functions.

Sculpting Tools

Brush

Tool to use for any of the *Sculpt* mode brushes.

Gesture Tools



General gesture tools to apply an operation via box, lasso, line and polyline shapes. See Gesture Tools for more information.

Mask Gesture Tools

Create a mask via a gesture.

Hide Gesture Tools

Hides/Shows geometry via a gesture.

Face Set Gesture Tools

Create a face set via a gesture.

Trim Gesture Tools

Perform a Boolean operation via a gesture.

Line Project

Flatten the geometry towards a drawn line.

Filter Tools

Tools for applying effects on the entire unmasked and visible mesh.

Mesh Filter
Cloth Filter
Color Filter

Mesh Filter

Apply a deformation to all unmasked vertices.

Cloth Filter

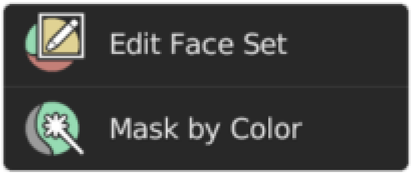
Applies a cloth simulation to all unmasked vertices.

Color Filter

Changes the active color attribute on all unmasked vertices.

Single Click Tools

Simpler tools that apply an operation on surfaces that are clicked on.



Edit Face Set

Modifies the face set under the cursor.

Mask by Color

Create a mask from any color from the color attribute by clicking on it.

General Tools

General transform and annotate tools like in other modes.

Move

Translation tool.

Rotate

Rotation tool.

Scale

Scale tool.

Transform

Adjust the objects translation, rotations and scale.

Annotate

Draw free-hand annotation.

Annotate Line

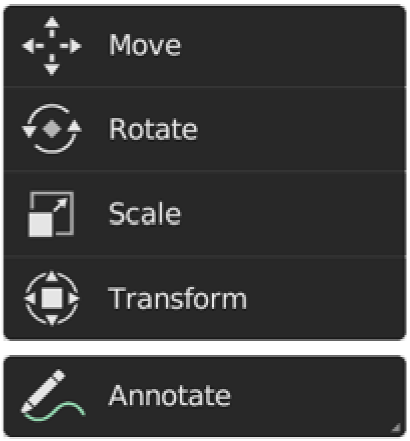
Draw straight line annotation.

Annotate Polygon

Draw a polygon annotation.

Annotate Eraser

Erase previous drawn annotations.


Move
Rotate
Scale
Transform
Annotate

Tools

- Brush
- Mask Gesture Tools
- Hide Gesture Tools
- Face Set Gesture Tools
- Trim Gesture Tools
- Line Project
- Mesh Filter
- Cloth Filter
- Color Filter
- Edit Face Set
- Mask by Color
- Transforms

Brush

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Brush*

Tool to use for any of the *Sculpt* mode brushes. Activating a brush from an asset shelf or brush selector will also activate this tool for convenience.

See the list of Essentials Brushes (based on available Brush Types) fore more details.

Mask Gesture Tools

Reference

Mode::	Sculpt Mode

Mask gesture tools apply a constant value to all selected vertices within the selection area. By default, these tools fully mask each vertex. Holding `Ctrl` while performing the selection clears the mask.

All mask gesture tools can be activated in the Toolbar and are comprised of the following:

Box Mask

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Box Mask*
Shortcut::	`B`

Creates a new Mask based on a box gesture.

Lasso Mask

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Lasso Mask*

Creates a new Mask based on a lasso gesture.

Line Mask

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Line Mask*

Creates a new Mask based on a line gesture.

Polyline Mask

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Polyline Mask*

Creates a new Mask based on a polyline gesture.

Tool Settings

Front Faces Only
: Only creates a mask on the faces that face towards the view.

Hide Gesture Tools

Reference

Mode:: Sculpt Mode

Hide gesture tools hide all selected vertices within the selection area and any of their connected edges and faces. Holding `Ctrl` while performing the selection reveals the vertices, edges, and faces.

Pressing `LMB` with any of these tools without also dragging reveals all elements of a mesh.

All hide gesture tools can be activated in the Toolbar and are comprised of the following:

Box Hide

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Box Hide*

Hides vertices and connected edges and faces based on a box gesture.

Lasso Hide

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Lasso Hide*

Hides vertices and connected edges and faces based on a lasso gesture.

Line Hide

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ‣ *Line Mask*

Hides vertices and connected edges and faces based on a line gesture.

Polyline Hide

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ‣ *Polyline Mask*

Hides vertices and connected edges and faces based on a polyline gesture.

Note

The Polyline Hide tool does not support showing all vertices via pressing `LMB`.

Tool Settings

Visibility Area

Determines whether all vertices inside or outside the selected area should be affected.

Inside::	All vertices and connected elements inside the selection area will be hidden.
Outside::	All vertices and connected elements outside the selection area will be hidden.

Face Set Gesture Tools

Reference

Mode:: Sculpt Mode

Face Set gesture tools apply a single new Face Set to all faces within the selection area.

All Face Set gesture tools can be activated in the Toolbar and are comprised of the following:

Box Face Set

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Box Face Set*

Creates a new Face Set based on a box gesture.

Lasso Face Set

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Lasso Face Set*

Creates a new Face Set based on a lasso gesture.

Line Face Set

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Line Face Set*

Creates a new Face Set based on a line gesture.

Polyline Face Set

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Polyline Face Set*

Creates a new Face Set based on a polyline gesture.

Tool Settings

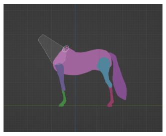
Front Faces Only
: Only creates a face set on the faces that face towards the view.

Trim Gesture Tools

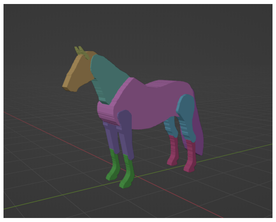
Reference

Mode:: Sculpt Mode

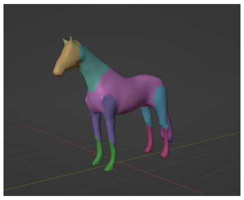
Trim gesture tools add or remove geometry based on a selection area. This tool is especially useful for sketching an early base mesh for further sculpting with the voxel remesher.



Using Lasso Trim set to *Join*



The symmetrized mesh.



Sculpting with voxel remeshing.

New geometry is assigned to a new Face Set. When removing geometry, the new interior geometry along the selection will be assigned a new face set instead.

Note

It is not recommended to use this tool on a mesh above 100k vertices when using *Difference* or *Union* as the Trim Mode with the *Exact* Solver. This tool is using a Boolean operation so it might take a long time to process. For higher resolution meshes it is recommended to instead use the Line Project tool or the *Fair Positions* mode of the Edit Face Set tool to trim geometry.

Box Trim

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Box Trim*

Performs a Boolean operation based on the area defined by a box gesture.

Lasso Trim

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Lasso Trim*

Performs a Boolean operation based on the area defined by a lasso gesture.

Line Trim

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Line Trim*

Performs a Boolean operation based on the area defined by a line gesture.

Note

The Line Trim tool does not support adding geometry. Only *Difference* mode is supported.

Polyline Trim

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ‣ *Polyline Trim*

Performs a Boolean operation based on the area defined by a polyline gesture.

Tool Settings

Solver

Algorithm used to calculate the Boolean intersections.

Fast::	Uses a mathematically simple solver which offers the best performance; however, this solver lacks support for overlapping geometry.
Exact::	Uses a mathematically complex solver which offers the best results and has full support for overlapping geometry; however, this solver is much slower than the *Fast Solver*.

Trim Mode

Geometry can be either added or removed by choosing one of these modes.

Difference::	Removes geometry, filling any holes that are created.
Union::	Creates a geometry and joins any intersections with existing geometry.
Join::	Similar to *Union* but joins the mesh as separate geometry, without performing any Boolean operations with existing geometry.

Shape Orientation

The method used to orientate the trimming shape.

View::	Use the view to orientate the trimming shape.

Surface::	Use the surface normal to orientate the trimming shape.

Extrude Mode

Fixed::	Aligns new geometry orthogonally for 90 degree angles in depth.
Project::	Aligns new geometry with the perspective of the current view for a tapered result.

Use Cursor for Depth

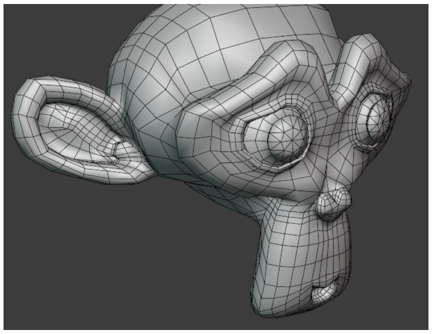
Use cursor location and radius for the dimensions and position of the trimming shape. If not set, the tool uses the full depth of the object from the camera view.

Line Project

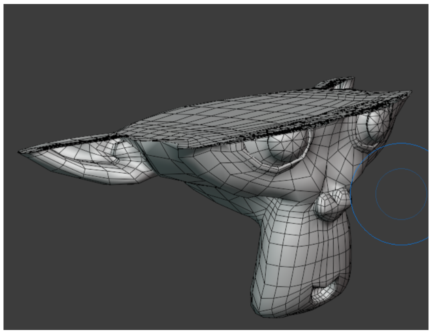
Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Line Project*

This tool flattens the geometry along a plane determined by the camera view and a drawn line. The region of the mesh being flattened is visualized by the side of the line that is shaded.



Before Line Project.



After Line Project.

Usage

Use the tool by:

1. Orient the 3D Viewport to define the direction in depth.
2. `LMB` and hold while moving the cursor to define direction of the line projection.
3. Adjust the operation with extra *Controls* shortcuts.
4. Release `LMB` to confirm.

Controls

Flip `F`
: Changes the side of the line that the tool projects geometry.

Snap `Ctrl`
: Constrains the rotation of the line to 15 degree intervals.

Move `Ctrl-Spacebar`
: Reposition the line.

Tool Settings

Limit to Segment
: The affected area will not extend the length of the drawn line. This helps defining a smaller area instead of extending the line infinitely long

Mesh Filter

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Mesh Filter*

Applies a deformation to all vertices in the mesh at the same time. Masking, auto-masking and visibility will be taken into account.

To use this tool, click and drag away from left to right or from right to left for a negative effect.

Tool Settings

Filter Type

These are all of the available filter deformations.

Smooth::	Smooths the positions of the vertices to either polish surfaces or remove volume from larger shapes. Especially useful to fix most of the artifacts of the voxel remesher. This filter works similar to the Smooth brush.
Scale::	Increases the size of the mesh. This filter works similar to the Scale Transform.
Inflate::	Displaces vertices uniformly along their normal. This filter works similar to the Inflate brush.
Sphere::	Morphs the mesh progressively into a sphere. This filter works similar to the To Sphere Transform.
Random::	

	Randomly moves vertices along the vertex normal. This filter works similar to the Randomize Transform.
Relax::	Tries to create an even distribution of quads without deforming the volume of the mesh. This filter works the same as holding `Shift` with the Slide Relax brush.
Relax Face Sets::	This will remove the jagged lines visible after drawing or creating a face set. This filter works the same as holding `Shift` with the Draw Face Set brush.
Surface Smooth::	Eliminates irregularities of the mesh by making the positions of the vertices more uniform while preserving the volume of the object. This filter works similar to the *Surface* deformation type of the Smooth brush. Shape Preservation How much of the original shape is preserved when smoothing. Per-Vertex Displacement How much the position of each individual vertex influences the final result.
Sharpen::	Sharpens and smooths the mesh based on its curvature, resulting in pinching hard edges and polishing flat surfaces. Especially useful when sculpting hard surfaces and stylized models with creasing and flattening brushes. Smooth Ratio How much smoothing is applied to polished surfaces. Intensify Details Increases the high frequency surface details of the mesh by intensifying the difference between creases and valleys. Curvature Smooth Iterations

The number of times the smoothing operation is applied per brush step. Controls how much smooth the resulting shape is, ignoring high-frequency details.

Enhance Details:: Increases the high frequency surface details of the mesh by intensifying the difference between creases and valleys. This filter works similar to the inverted direction of the Smooth brush.

Erase Displacement:: Deletes displacement information of the Multires Modifier, resetting the mesh to a regular subdivision surface result. This can be used to reset parts of the sculpt or to fix reprojection artifacts after applying a Shrinkwrap Modifier. Negative strokes will intensify the displacement details, this method works similar to *Enhance Details* and can give better results in some circumstances.

Strength

The amount of effect the filter has on the mesh. At certain object scales it can be useful to change this value.

Deformation Axis

Apply the deformation only on the selected axis.

Orientation

Orientation of the axis to limit the filter displacement.

Local::	Use the local axis to limit the displacement.
World::	Use the global axis to limit the displacement.
View::	Use the view axis to limit the displacement.

Cloth Filter

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ‣ *Cloth Filter*

This tool works similar to the Cloth Brush, however, it applies a cloth simulation to all vertices in the mesh at the same time. Click and drag away from the object for a positive effect and towards for a negative effect.

Tip

Vertices can be "pinned" by masking vertices that should remain stationary, or by using Face Sets.

Tool Settings

Filter Type

Operation that is going to be applied to the mesh.

Gravity::	Applies gravity to the simulation.
Inflate::	Inflates the cloth.
Expand::	Expands the cloth's dimensions.
Pinch::	Pinches the cloth to the point where the cursor was when the filter started.
Scale::	Scales the mesh as a Soft Body using the distance to the origin of the object as scale. This creates filter produces folds in the surface. The orientation of the folds can be controlled using the *Force Axis* and *Orientation*.

Strength

The amount of effect the filter has on the mesh.

Force Axis
Apply the force along the selected axis.

Orientation
Orientation of the axis to limit the filter force.

Local::	Use the local axis to limit the force and set the gravity direction.
World::	Use the world axis to limit the force and set the gravity direction.
View::	Use the view axis to limit the force and set the gravity direction.

Cloth Mass
Mass of each simulation particle.

Cloth Damping
How much the applied forces are propagated through the cloth.

Use Face Sets
Only applies the cloth forces to the vertices assigned to the Face Set that are under the mouse.

Use Collisions
Enables the detection of collisions with other objects during the simulation. In order for the sculpt object to collide with object, the collision object must have Collision Physics activated.

Color Filter

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ▸ *Color Filter*

Apply color corrections or effects on the active color attribute on all vertices in the mesh at the same time.

To use this tool, click and drag away from left to right or from right to left for a negative effect.

Tool Settings

Filter Type

Fill::	Fills in a single color.
Hue::	Shifts the Hue of each color.
Saturation::	Increases or decreases the saturation.
Value::	Increases or decreases the values.
Brightness::	Increases or decreases the brightness.
Contrast::	Increases or decreases the contrast.
Smooth::	Blurs or sharpens the colors.
Red::	Increases or decreases the red channel.
Green::	Increases or decreases the green channel.
Blue::	Increases or decreases the blue channel.

Fill Color

Set a color that will be used for the fill filter type.

Strength

The amount of effect the filter has on the color attribute.

Edit Face Set

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Edit Face Set*
Operator::	Grow/Shrink Face Sets

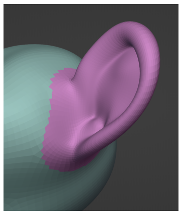
Edits the Face Set under the cursor.

Tool Settings

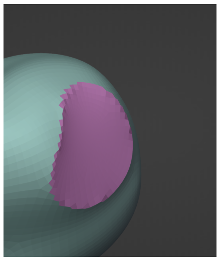
Mode

The operation to apply to the face set.

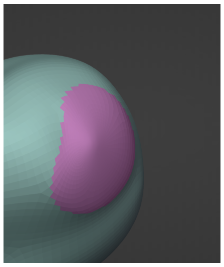
Grow Face Set::	Grows the face sets boundary by one face based on mesh topology. This is also available as a shortcut operator via `Ctrl-W`.
Shrink Face Set::	Shrinks the face sets boundary by one face based on mesh topology. This is also available as a shortcut operator via `Ctrl-Alt-W`.
Delete Geometry::	Deletes the faces that are assigned to the face set.
Fair Positions::	Creates a perfectly flat and smooth geometry patch from the face set. This is the ideal way to trim parts of your mesh if the vertex count is too high for other operations, or the vertex IDs must not be altered (Like when using Multires sculpting).
Fair Tangency::	Creates a smooth as possible geometry patch from the face set by minimizing changes in vertex tangents. This is ideal for creating smooth curved surfaces on complex topology, where just using the smooth brush will not lead to desired results



Before fairing.



After using Fair Positions.



After using Fair Tangency.

Strength

The amount of effect the filter has on the mesh. This setting is only available for the fairing operations.

Modify Hidden

Apply the edit operation to hidden face sets.

Mask by Color

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Mask by Color*

Click on any color on the mesh to create a new mask (based on the active color attribute).

Tool Settings

Threshold

How much changes in color affect the mask generation. A smaller threshold includes fewer similar colors. A larger threshold includes much more similar colors.

Contiguous

Mask only contiguous color areas. Colors that don't touch the one that you click on will not be masked.

Invert

Invert the generated mask.

Preserve Previous Mask

Preserve previous mask and add or subtract the new one generated by the colors.

Transforms

Move

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ▸ Move*

Translation tool.

Rotate

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ▸ Rotate*

Rotation tool.

Scale

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ▸ Scale*

Scale tool.

Transform

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar ‣ Transform*

Tool to adjust the objects translation, rotations and scale.

Tool Settings

Each tool has the following settings to change how the unmasked mesh will be transformed.

Transform Mode

How the transformation is going to be applied to the target.

All Vertices::	Applies the transformation to all vertices in the mesh.
Elastic::	Applies the transformation while dynamically simulating elasticity. Instead of applying this to all vertices, it uses the radius of the cursor as the area of effect.

Tool Settings

- Brushes
    - Introduction
    - Manage Brushes
    - Brush Settings
- Brush Settings
- Dyntopo
- Remesh
    - Known Limitations
- Symmetry
- Options
    - Gravity

Brush Settings

Information on brush settings for every mode can be found in these pages:

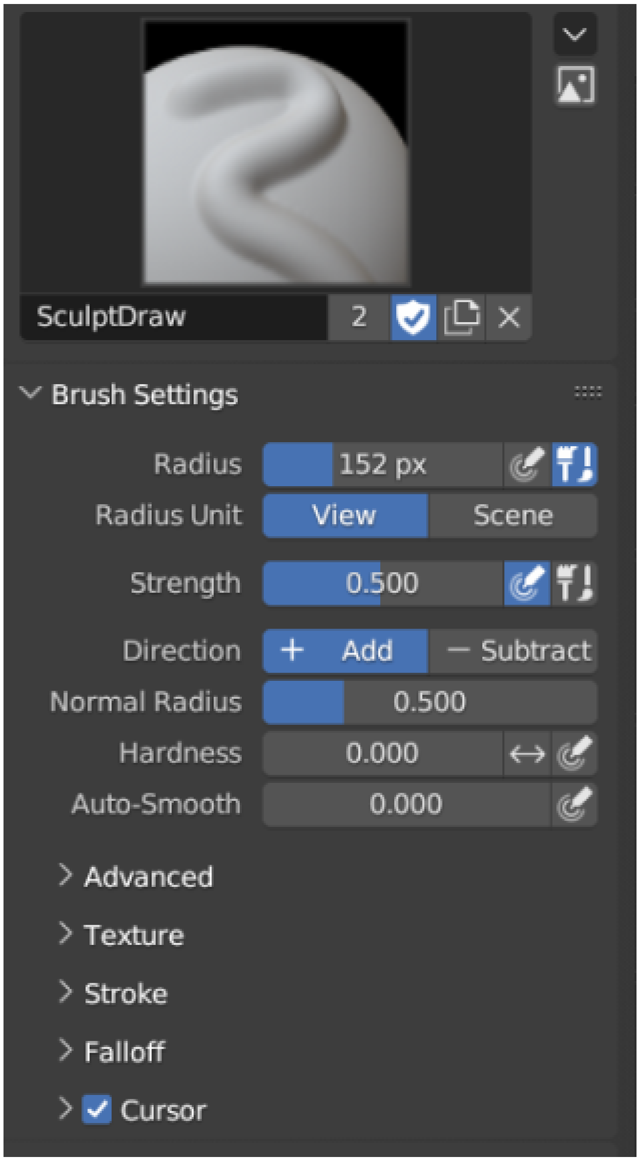
Brush
: General and advanced settings.

Texture
: Color and mask texture settings.

Stroke
: Stroke methods and settings.

Falloff
: Falloff curves and settings.

Cursor
: Cursor and appearance settings.



SculptDraw 2

Radius: 152 px
Radius Unit: View
Strength: 0.500
Direction: Add
Normal Radius: 0.500
Hardness: 0.000
Auto-Smooth: 0.000
☑ Cursor

Brush Settings
View
Scene
+ Add
− Subtract
Advanced
Texture
Stroke
Falloff

Brush settings panel.

Dyntopo

Reference

Mode::	Sculpt Mode
Panel::	*Sidebar ‣ Tool ‣ Dyntopo*

Dynamic Topology (aka Dyntopo) can be toggled with the checkbox. With dynamic topology active, most brushes will subdivide the mesh during the stroke.

For a general explanation of dynamic topology, visit the Introduction.

Detail Size/Percentage, Resolution `R`

Each Detail Type's detail is set here. Depending on the Detail Type being used this property will rather show as a pixel count (px), or percentage.

`R` allows you to interactively set the detail, with a preview of the detail's density in the 3D Viewport.

Sample Detail Size (pipette icon)

When using *Constant Detail*, it is possible to sample the detail value of a certain mesh area by clicking the pipette icon next to the detail setting and then clicking on the area.

Refine Method

Setting the option will determine which of the methods will be used when altering the topology.

Subdivide Edges::	Just like the Subdivide tool, this method will only subdivide topology to match the detail given.
Collapse Edges::	When topology is too dense, and is smaller than the detail given, edges will be collapsed to fit the detail size appropriately.

Subdivide Collapse::	This method combines the two methods, subdividing edges smaller than the detail size, and collapsing topology.

Detailing

Dyntopo uses the following different detail methods to create dynamic detail on an object.

Relative Detail::	This method uses a detail size based on the number of pixels, and in turn will create topology in that size. Zoom out big details, zoom in small fine details.
Constant Detail::	To keep detail uniform across the entire object, Constant Detail can be used. The detail is based on the percentage of a single unit.
Brush Detail::	Giving more control over the topology, with this method you can create topology based on the brush size. You can increase and lower topology by resizing the brush itself. The detail size is based the size of the brush itself, where full detail will create topology the size of the brush radius itself.
Manual Detail::	Similar to constant detail, this value sets a percentage value uniform across the object, but only applies detailing changes when using Flood Fill.

Detail Flood Fill `Ctrl-R`

When using *Constant* or *Manual Detailing*, this option is made available, allowing you to fill the entire object with a uniform detail, based on the detail size.

Remesh

Reference

Mode::	Sculpt Mode
Header::	*Tool Settings ‣ Remesh*
Panel::	*Sidebar ‣ Tool ‣ Remesh*
Shortcut::	`Ctrl-R`

For a general explanation to remeshing, visit the Introduction.

Voxel Size `R`

The resolution or the amount of detail the remeshed mesh will have. The value is used to define the size, in object space, of the Voxel. These voxels are assembled around the mesh and are used to determine the new geometry. For example a value of 0.5m will create topological patches that are about 0.5m (assuming *Preserve Volume* is enabled). Lower values preserve finer details but will result in a mesh with a much more dense topology.

The voxel size also be adjusted from the 3D Viewport using `R`. Using the shortcut displays an interactive grid overlay showing the resulting voxel size. Moving the mouse closer to center of the grid decreases the voxel size while moving away from the center increase the voxel size. Holding `Shift` increases the precision; adjusting the voxel size in small increments.

Sample Voxel Size

Used to adjust the *Voxel Size* by picking an area of the mesh to match the denseness of polygons after the remesh operation.

Adaptivity

Reduces the final face count by simplifying geometry where detail is not needed. This introduce triangulation to faces that do not need as much detail. Note, an *Adaptivity* value greater than zero disables *Fix Poles*.

Fix Poles

Tries to produce less poles at the cost of some performance to produce a better topological flow.

Preserve

Volume

Tells the algorithm to try to preserve the original volume of the mesh. Enabling this could make the operator slower depending on the complexity of the mesh.

Paint Mask

Reprojects the paint mask onto the new mesh.

Face Sets

Reprojects Face Sets onto the new mesh.

Color Attributes

Reprojects the Color Attributes onto the new mesh.

Voxel Remesh

Performs the remeshing operation to create a new manifold mesh based on the volume of the current mesh. Performing this will lose all mesh object data layers associated with the original mesh.

See also

Remesh modifier

Known Limitations

- Remeshing only works on the original mesh data and ignores generated geometry from modifiers, shape keys, rigging, etc.
- Remeshing will not work with the Multiresolution Modifier.

Symmetry

Reference

Mode:: Sculpt Mode
Tool:: *Toolbar ‣ Tool ‣ Symmetry*

Mirror

Mirror the brush strokes across the selected local axes. Note that if you want to alter the directions the axes point in, you must rotate the model in Edit Mode and not in Object Mode.

Lock

These three buttons allow you to block any modification/deformation of your model along selected local axes, while you are sculpting it.

Tiling

Using this option allows you to seamlessly tile your strokes along the given axes. This allows to create repeating patterns.

Feather

Reduces the strength of the stroke where it overlaps the planes of symmetry.

Radial X, Y, Z

These settings allow for radial symmetry in the desired axes. The number determines how many times the stroke will be repeated within 360 degrees around the central axes.

Tile Offset X, Y, Z

The offset allows the option to alter the tile size along all three axes. The default tile size is set to one unit.

Symmetrize

Direction

Determines which direction the model will be symmetrized.

Merge Distance

A parameter of the *Symmetrize* operator to control the distance within which symmetrical vertices are merged.

Symmetrize

Uses direction orientation to symmetrize. Since Dyntopo adds details dynamically it may happen that the model becomes asymmetric, so this a good tool for that.

Options

Reference

Mode::	Sculpt Mode
Tool::	*Toolbar* ▸ *Options*

Display

Fast Navigate

For multiresolution models, shows low resolution while navigating in the viewport.

Delay Viewport Updated

Update the geometry when it enters view. This provides for faster navigation.

Use Deform Only

Limits the activated modifiers on the active object to Deform Modifiers, and Multiresolution. Constructive modifiers (like Subdivision Surface, Mirror and other) get deactivated, because they could give inaccurate results.

See also

See the Display options.

Gravity

Factor

Setting the factor allows you to add gravity to your brush strokes, giving it a draping effect.

Orientation

Using another object, the gravity can be oriented to the set object's local Z axis, changing the direction of the gravity.

Controls

Auto-Masking

Reference

Mode::	Sculpt Mode
Tool::	*Header* ‣ *Auto-Masking*
Shortcut::	`Alt-A`

These properties automatically mask geometry based on geometric features of the mesh. It's an quick alternative to frequent manual masking. These masks are initialized on every new stroke or tool usage. They are also never visible as an overlay.

Note, these properties are applied across all sculpt brushes, however, they can also be configured per brush in the Advanced Brush Settings.

These properties can be accessed via a Pie Menus by pressing `Alt-A`.

All auto-masking modes can be combined, which makes the generated auto-mask more specific. For example it's possible to auto-mask a specific face set, while excluding disconnected topology and face set boundaries, and only affect faces that are oriented towards the view via View Normal.

Topology

> Only vertices that are topologically connected to where you started the stroke/tool on are affected. So loose geometry islands will be auto-masked.
>
> Additionally for the Grab and Thumb brushes, anything that is not connected within the brush radius will be auto-masked. So even if geometry is connected somewhere, it is considered separate if the connection is not within the radius.

Face Sets

Only vertices that are part of the same face set that you started the stroke/tool on are affected.

Tip

If no topology or face set is visible under the cursor at the start of the stroke, the previously auto-masked area will be targeted. This is especially useful with the "Projected" falloff shape in the Falloff Settings.

Mesh Boundary

Vertices that are part of open boundary edges are not affected. This also includes boundary edges to hidden faces.

Propagation Steps

Increases the soft gradient towards the auto-masked boundary edges. Each step iterates the distance one edge further. This setting is used for both Mesh Boundary and Face Sets Boundary.

Create Mask

This will execute the Mask From Mesh Boundary operator with the current auto-masking settings. This is very useful to visualize the current auto-mask, or to edit the mask further manually.

Face Sets Boundary

Vertices that are part of a boundary between face sets are not affected. This also includes boundary edges to hidden faces. Propagation Steps are shared with Mesh Boundary auto-masking.

Create Mask

This will execute the Mask From Face Sets Boundary operator with the current auto-masking settings. This is very useful to visualize the current auto-mask, or to edit the mask further manually.

Cavity

Vertices that are the peaks of the surface curvature are not affected. While this auto-mask is primarily meant for painting, it can also be used for regular sculpting.

Factor

The overall contrast of how strong the cavity is applied. The value of 0.5 is the default, but better results can also be achieved on 0.2 if a Custom Curve is used as well.

Blur

The number of times the cavity mask is blurred. A value of 0 will give the pure cavity auto-mask. Anything higher than 6 will likely have a less visible effect and decrease performance. Even though the value is capped to 10, it can be increased up to 25 if typing in the value.

Custom Curve

Use a custom curve to fine tweak the cavity auto-mask. This is very useful if only small crevices or flat surfaces should be affected. Or for example if the contrast should be increased/decreased in a specific way.

Create Mask

This will execute the Mask From Cavity operator with the current auto-masking settings. This is very useful to visualize the current auto-mask, or to edit the mask further manually.

Cavity (Inverted)

This is the same as "Cavity", but inverted. This means the valleys/crevices of the surface curvature will not be affected. It cannot be used at the same time as Cavity and shares all of its settings. Enable this to quickly invert the cavity auto-mask.

View Normal

Only vertices with a Normal that face the viewer are affected. This is similar to the "Front Faces Only" toggle in the Brush Setting, to only affect visible geometry. The advantage of this auto-mask is that it has more options and works on sculpt mode as a whole.

Occlusion

Change the View Normal behavior to only affect vertices that are not occluded by other faces. This setting is incompatible with the other Limit and Falloff sliders. It also causes a much slower performance.

Limit

Determines the range of angles that will be affected. 90 degrees encompasses all that is visible.

Falloff

Extends the angular range of the Limit slider with a soft falloff gradient. This falloff will visually extend the limit range further.

Area Normal

Very similar to the View Normal, but uses the Normal of the surface that you started the stroke/tool on. This way any direction can be chosen for what vertices will be affected. It has the same Limit and Falloff sliders as the View Normal auto-mask.

Editing

- Sculpt
    - Transform
    - Show & Hide
    - Fairing
    - Trimming
    - Mesh Filters
    - Sample Color
    - Set Pivot
    - Rebuild BVH
    - Dynamic Topology Toggle
    - Transfer Sculpt Mode
- Mask
    - Invert Mask
    - Fill Mask
    - Clear Mask
    - Box Mask
    - Lasso Mask
    - Mask Filters
    - Expand Mask
    - Mask Extract
    - Mask Slice
    - Mask From Cavity
    - Mask From Mesh Boundary
    - Mask From Face Sets Boundary
    - Random Mask
    - Display Settings
    - Clear Sculpt-Mask Data
- Face Sets
    - Face Set from Masked
    - Face Set from Visible
    - Face Set from Edit Mode Selection
    - Initialize Face Sets
    - Grow/Shrink Face Sets

  - Expand Face Set
  - Extract Face Set
  - Randomize Colors
  - Display Settings
- Expand
  - Expand Mask by Topology
  - Expand Mask by Normals
  - Expand Face Set by Topology
  - Expand Active Face Set

Sculpt

This page details the general hotkey operators and menu operators in sculpt mode.

Transform

Move
: Change the position of the object.

Rotate
: Change the orientation of the mesh.

Scale
: Increase/decrease the size of the mesh.

Sphere
: Morph the mesh to a spherical shape.

See also

Transform Tools.

Show & Hide

Reference

Mode::	Sculpt Mode
Menu::	*Sculpt*

Some very common hotkey operators to control the visibility based on face sets. These are not part of any menu and have to be used via the shortcuts.

More visibility operators can be found in the Face Sets Menu and the Pie Menu shortcut `W`. (Since visibility is often toggled via face sets.)

Box Hide
  Draw a box to hide faces of a mesh.

Box Show
  Draw a box to reveal hidden faces. This works similar to the Box Select tool.

Toggle Visibility `Shift-H`
  Hide all face sets except the active one (under the cursor). If face sets are already hidden, then this operator will show everything.

Hide Active Face Set `H`
  Hide the face set under the cursor. Press `Shift-H` afterwards to show everything.

Show All `W`, `Alt-H`
  Reveal all hidden geometry.

Invert Visible
  Hides all visible geometry and makes all hidden geometry visible.

Hide Masked
  Hides all masked vertices.

Grow/Shrink Visibility `PageUp`, `PageDown`
  Grows or shrinks the visible area of the mesh along its surface.

See also

For a more general introduction see Visibility, Masking & Face Sets.

Fairing

These operators smooths geometry patches based of a Face set.

See also

Edit Face Set Tool

Fair Positions

Creates a perfectly flat and smooth geometry patch from the face set. This is the ideal way to trim parts of your mesh if the vertex count is too high for other operations, or the vertex IDs must not be altered (Like when using Multires sculpting).

Fair Tangency

Creates a smooth as possible geometry patch from the face set by minimizing changes in vertex tangents. This is ideal for creating smooth curved surfaces on complex topology, where just using the smooth brush will not lead to desired results

Trimming

The trimming operators add or remove geometry from the mesh based on a gesture input. These operators are especially useful for sketching an early base mesh for further sculpting with the voxel remesher.

Line Project

Flattens the geometry along a plane determined by the camera view and a drawn line. The region of the mesh being flattened is visualized by the side of the line that is shaded.

Box Trim

Removes geometry based on a box selection.

Lasso Trim

Removes geometry based on a lasso selection.

Box Add

Adds geometry based on a box selection.

Lasso Add
: Adds geometry based on a lasso selection.

Mesh Filters

Applies a deformation to all vertices in the mesh at the same time. Masking, auto-masking and visibility will be taken into account.

To use these operators, click and drag away from left to right or from right to left for a negative effect.

See also

Mesh Filter Tool

Smooth
: Smooths the positions of the vertices to either polish surfaces or remove volume from larger shapes. Especially useful to fix most of the artifacts of the voxel remesher. This filter works similar to the Smooth brush.

Surface Smooth
: Eliminates irregularities of the mesh by making the positions of the vertices more uniform while preserving the volume of the object. This filter works similar to the *Surface* deformation type of the Smooth brush.

Inflate
: Displaces vertices uniformly along their normal. This filter works similar to the Inflate brush.

Relax Topology
: Tries to create an even distribution of quads without deforming the volume of the mesh. This filter works the same as holding `Shift` with the Slide Relax brush.

Relax Face Sets

This will remove the jagged lines visible after drawing or creating a face set. This filter works the same as holding `Shift` with the Draw Face Set brush.

Sharpen

Sharpens and smooths the mesh based on its curvature, resulting in pinching hard edges and polishing flat surfaces. Especially useful when sculpting hard surfaces and stylized models with creasing and flattening brushes.

Enhance Details

Increases the high frequency surface details of the mesh by intensifying the difference between creases and valleys. This filter works similar to the inverted direction of the Smooth brush.

Erase Multires Displacement

Deletes displacement information of the Multires Modifier, resetting the mesh to a regular subdivision surface result. This can be used to reset parts of the sculpt or to fix reprojection artifacts after applying a Shrinkwrap Modifier.

Negative strokes will intensify the displacement details, this method works similar to *Enhance Details* and can give better results in some circumstances.

Randomize

Randomly moves vertices along the vertex normal. This filter works similar to the Randomize Transform.

Sample Color

Reference

Mode::	Sculpt Mode
Menu::	*Sculpt ‣ Sample Color*

Shortcut:: `Shift-X`

Adjust the brush color of the Paint tool to the color under the mouse cursor.

Set Pivot

Reference

Mode::	Sculpt Mode
Menu::	*Sculpt ‣ Set Pivot*

Like Object and Edit Mode, Sculpt Mode also has a Pivot Point. This is because the basic move, rotate and scale transforms are also supported in Sculpt Mode. But the pivot point in Sculpt Mode is unique. It always moves together with the transformed mesh and can be both manually & automatically placed.

Origin
: Sets the pivot to the origin of the sculpt.

Unmasked
: Sets the pivot position to the average position of the unmasked vertices.

Mask Border
: Sets the pivot position to the center of the mask's border. This operation will automatically happen when using Expand.

Active Vertex
: Sets the pivot position to the active vertex position.

Surface `Shift-RMB`
: Sets the pivot position to the surface under the cursor.

Tip

For more convenient placement of the pivot point it’s recommended to use the shortcut assigned to *Surface*.

See also

For a more general introduction see Transforming.

Rebuild BVH

Reference

Mode:: Sculpt Mode
Menu:: *Sculpt ‣ Rebuild BVH*

Recalculates the BVH used by Dyntopo to improve performance, which might degrade over time while using Dyntopo.

See also

For a more general introduction see Adaptive Resolution.

Dynamic Topology Toggle

Toggles Dyntopo.

Transfer Sculpt Mode

Reference

Mode:: Sculpt Mode
Menu:: *Sculpt ‣ Transfer Sculpt Mode*
Shortcut::

`Alt-Q`

Switches Sculpt Mode from the Active object to the object under the mouse. See Switching Objects for more information.

See also

For a more general introduction see Working with Multiple Objects.

Mask

This page details the mask related shortcut operators and menu operators in sculpt mode. Other related information to masks can also be found at the bottom of the page.

Reference

Mode::	Sculpt Mode
Menu::	*Mask*
Shortcut::	`A`

Masks can be edited across all visible faces. Using `A` opens a pie menu to choose the most common operations.

Invert Mask

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Invert Mask*
Shortcut::	`A`, `Ctrl-I`

Inverts the visible mask. This is often useful when the masked vertices are the surfaces you want to sculpt/paint. In that case create a mask and then invert it.

Fill Mask

Reference

Mode::	Sculpt Mode
Menu::	

Mask ‣ Fill Mask

Fully masks the visible geometry. Alternatively it is common to clear and then invert a mask via `A` to achieve the same effect.

Clear Mask

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Clear Mask*
Shortcut::	`A`, `Alt-M`

Removes the mask on all visible vertices. To completely remove the mask data, see Clear Sculpt-Mask Data.

Box Mask

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Box Mask*
Shortcut::	`B`

Works like the Box Mask tool, it creates a rectangular mask region. Hold `Shift` or press `MMB` to clear the mask of the selected region.

Lasso Mask

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Lasso Mask*
Shortcut::	`Ctrl-RMB`

Can be used to create a free-form mask, similar to the Lasso Mask tool. This is very commonly used.

Tip

To clear the mask of areas with the *Lasso Mask*, first invert the mask, use *Lasso Mask*, and then invert the mask back.

Mask Filters

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Smooth/Sharpen Mask, Grow/Shrink Mask, Increase/Decrease Contrast*
Shortcut::	`A`

Similarly to other Filter Tools, mask filters are operations that are applied to the whole mask.

Type

Smooth/Sharpen Mask

Changes the sharpness of the mask edge. Using this can be faster and more consistent than smoothing the mask with the Mask brush.

Grow/Shrink Mask

Further grow or shrink the mask along the surface of the mesh.

Increase/Decrease Contrast

Changes the contrast of the mask.

In the Adjust Last Operation panel there are further options to add iterations for a stronger effect.

Iterations

The number of times the filter is applied.

Auto Iteration Count
Use an automatic number of iterations based on the number of vertices of the sculpt. Disable this option to set the Iterations manually.

Tip

An alternative to Iterations is to use Repeat Last via the shortcut `Shift-R`.

Expand Mask

Note

More info on Mask Expand along Topology at the Expand page.

Mask Extract

Reference

Mode::	Sculpt Mode
Menu::	*Mask* ‣ *Mask Extract*

Creates a duplicate mesh object based on masked geometry. The extracted geometry is also further processed by default for a cleaner result.

Threshold
Minimum mask value to consider the vertex valid to extract a face from the original mesh.

Add Boundary Loop
Creates and extra boundary loop on the edges of the geometry, making it easier smooth the boundaries and apply additional modifiers.

Smooth Iterations
Smooth iterations applied to the extracted mesh.

Project to Sculpt
: Project the extracted mesh on to the original sculpt object.

Extract as Solid
: Adds a Solidify Modifier to the newly created mesh object.

Mask Slice

Reference

Mode::	Sculpt Mode
Menu::	*Mask ▸ Mask Slice*

Removes the masked vertices from the mesh.

Threshold
: Minimum mask value to consider the vertex valid to extract a face from the original mesh.

Fill Holes
: Fills concave holes with geometry that might have resulted from the *Mask Slice* operation.

> **Tip**
>
> If nothing is masked, this operation can be used to just fill all holes. Especially when using Trim Tools tools and the Voxel Remesher

Slice to New Object
: Create a new object from the masked geometry.

Mask From Cavity

Reference

Mode:: Sculpt Mode
Menu:: *Mask ‣ Mask from Cavity*

Generates a mask based on the cavity of the surface. The settings of the operation can be changed in the Adjust Last Operation panel.

Mode
: Choose how the newly created mask is mixed with the existing one. By default it will replace the old mask via “Mix”.

Mix Factor
: The factor of the mix effect. Choose how strong the new mask is applied on the existing one.

Automask Settings
: The same settings as the Auto-Masking settings are applied.

Factor
: Same as Auto-Masking.

Blur
: Same as Auto-Masking.

Invert
: Same as Auto-Masking.

Custom Curve
: Same as Auto-Masking.

Mask From Mesh Boundary

Reference

Mode:: Sculpt Mode
Menu:: *Mask ‣ Mask from Mesh Boundary*

Generates a mask based on the topological islands of the mesh. The settings of the operation can be changed in the Adjust Last Operation panel.

Mode
    Choose how the newly created mask is mixed with the existing one. By default it will replace the old mask via “Mix”.

Mix Factor
    The factor of the mix effect. Choose how strong the new mask is applied on the existing one.

Automask Settings
    The same settings as the Auto-Masking settings are applied.

Propagation Steps
    Same as Auto-Masking.

Mask From Face Sets Boundary

Reference

Mode::	Sculpt Mode
Menu::	*Mask* ‣ *Mask from Face Sets Boundary*

Generates a mask based on the face set islands of the mesh. The settings of the operation can be changed in the Adjust Last Operation panel.

Mode
    Choose how the newly created mask is mixed with the existing one. By default it will replace the old mask via “Mix”.

Mix Factor
    The factor of the mix effect. Choose how strong the new mask is applied on the existing one.

Automask Settings
    The same settings as the Auto-Masking settings are applied.

Propagation Steps
   Same as Auto-Masking.

Random Mask

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Random Mask*

Generates a mask with random values for the entire object based on different mesh data.

Per Vertex
   Assigns a random mask value for each vertex.

Per Face Set
   Assigns a random mask value for each Face Set.

Per Loose Mask
   Assigns a random mask value for each disjoint part of the mesh.

Display Settings

Reference

Mode::	Sculpt Mode
Popover::	*Viewport Overlays – Sculpt ‣ Mask*

The mask display can be toggled as a viewport overlay. In the overlay popover, the opacity of the mask overlay can be adjusted to make it more or less visible on the mesh.

Clear Sculpt-Mask Data

Reference

Mode::	Object/Edit Mode
Menu::	*Properties* ‣ *Object Data* ‣ *Geometry Data* ‣ *Clear Sculpt-Mask Data*

Completely frees the mask data layer from the mesh. While not a huge benefit, this can speed-up sculpting if the mask is no longer being used.

Face Sets

This page details the face set related hotkey operators and menu operators in sculpt mode.

Tip

There is a face set pie menu that can be accessed with `Alt-W`.

Face Set from Masked

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Face Set from Masked*

Creates a new face set from Masked Geometry.

Face Set from Visible

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Face Set from Visible*

Creates a new face set from all visible geometry.

Face Set from Edit Mode Selection

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ▸ Face Set from Edit Mode Selection*

Creates a new face set corresponding to the Edit Mode face selection.

Initialize Face Sets

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ▸ Initialize Face Sets*

Initializes all face sets on the mesh at once based off one of several mesh attribute properties.

Mode

The mesh data attribute used to define the boundaries for the face sets.

By Loose Parts::	Creates a new face set per discontinuous part of the mesh.
By Face Set Boundaries::	Creates a face set for each isolated face set. This mode is useful for splitting the patterns created by Face Set Expand into individual Face Sets for further editing.
By Materials::	Creates a face set per Material Slot.
By Normals::	Creates face sets for Faces that have similar Normals.
By UV Seams::	Creates face sets using UV Seams as boundaries.
By Edge Creases::	Creates face sets using Edge Creases as boundaries.
By Edge Bevel Weight::	Creates face sets using Bevel Weights as boundaries.
By Sharp Edges::	Creates face sets using Sharp Edges as boundaries.

Threshold

The minimum value to consider a certain attribute a boundary when creating the face sets.

Grow/Shrink Face Sets

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Grow/Shrink Face Sets*
Tool::	Edit Face Set
Shortcut::	`Ctrl-W`, `Ctrl-Alt-W`

Expands or contracts the face set under the cursor by adding or removing surrounding faces.

Expand Face Set

Note

More info on Face Set Expand at the Expand page.

Extract Face Set

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Extract Face Set*

Creates a new mesh based on the selected face set. Once the operator is initiated, hover over the face set and `LMB` to create the new mesh. After the operator is finished the new mesh will be selected in Object Mode.

Randomize Colors

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Randomize Colors*

Generates a new set of random colors to render the face sets in the 3D Viewport.

Display Settings

Reference

Mode::	Sculpt Mode
Popover::	*Viewport Overlays – Sculpt ‣ Face Sets*

The face sets display can be toggled as a viewport overlay. In the overlay popover, the opacity of the face sets overlay can be adjusted to make it more or less visible on the mesh.

Expand

This is a multi-purpose modal operator to intuitively create and edit masks/face sets. When executed, it uniformly expand outwards a pattern from the vertex under the cursor.

Note

These operators are meant to be used interactively through the shortcut.

There is also a full showcase of the Expand features and use cases.

A preview of *Expand Mask by Topology*

Expand Mask by Topology

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Expand Mask by Topology*
Shortcut::	`Shift-A`

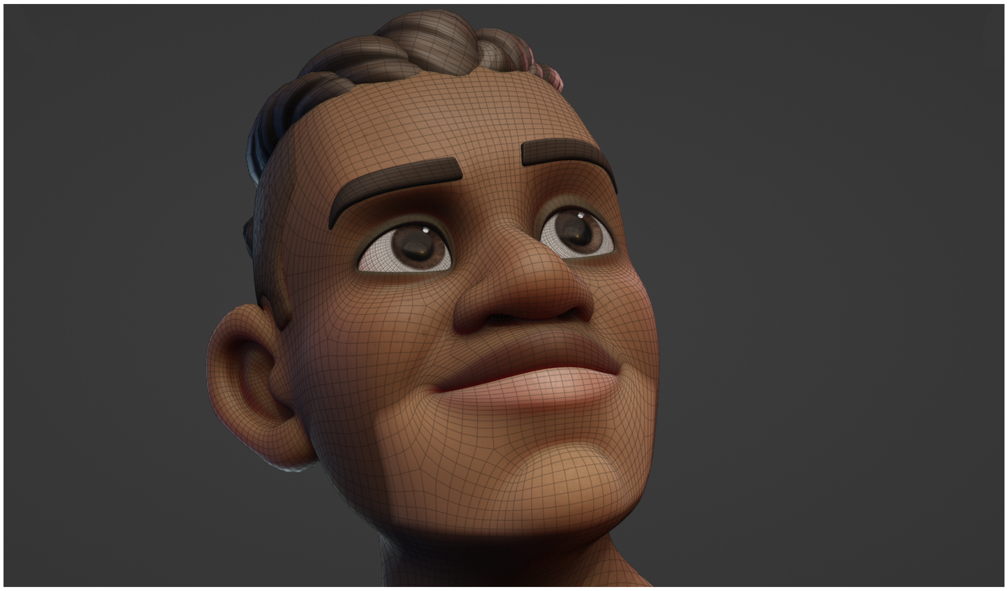
Expands a mask from the active vertex.

Usage

Mask from A to B

Start the operator and expand a mask from an origin to your mouse cursor distance. Then confirm with `LMB` or `Return`

By default the expansion will use a *Geodesic* falloff `1` to create perfectly accurate distances along the surfaces. Use other falloff types via `2`, `3` & `4` to expand via triangles, whole faces or scene distances instead.



The typical result when using the *Diagonals* falloff to expand along the quads of the face.

Start Expand while pointing at an open boundary to expand from the entire boundary loop. This will always use the *Topology* falloff mode.

Fill Connected Meshes

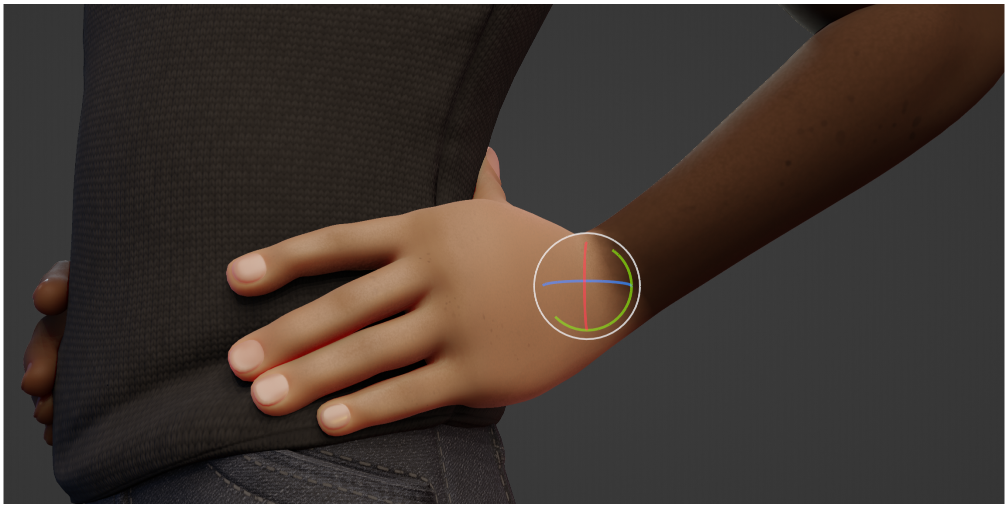
Move your cursor outside of the boundaries of the mesh to mask the entire connected mesh. This can be done repeatedly to quickly mask separate meshes.

Fill Face Sets

Hold `Ctrl` to snap to face sets under your cursor and fill them. Any face set that was already covered in the expansion will be filled as well.

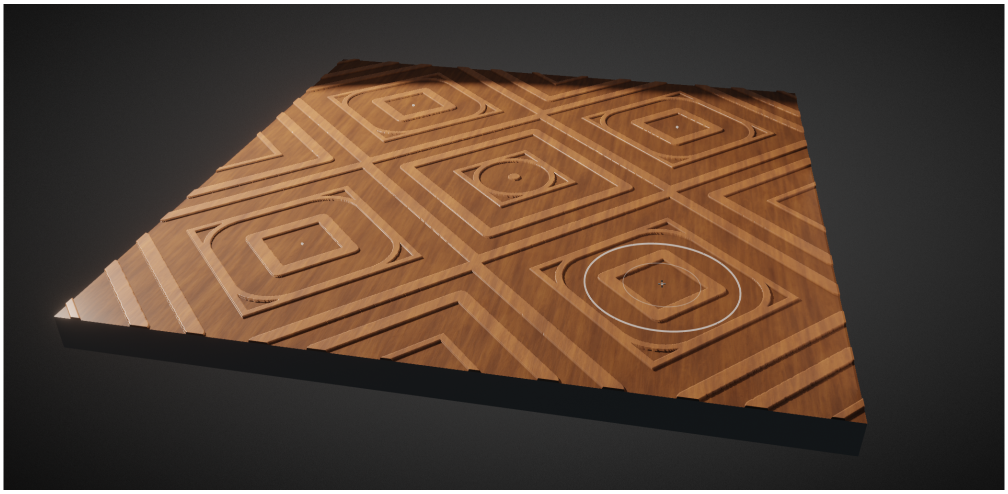
Automatically Set Transform Pivot

While using any Transform tool, the pivot point will automatically snap the border of an Expand result. This way areas (Like limbs) can be masked and then immediately rotated or otherwise transformed.



Pattern Creation

The different falloff types can be used for circular, triangular and square patterns. More loops can also be added/removed via `W` & `Q` to repeat the pattern across the mesh.



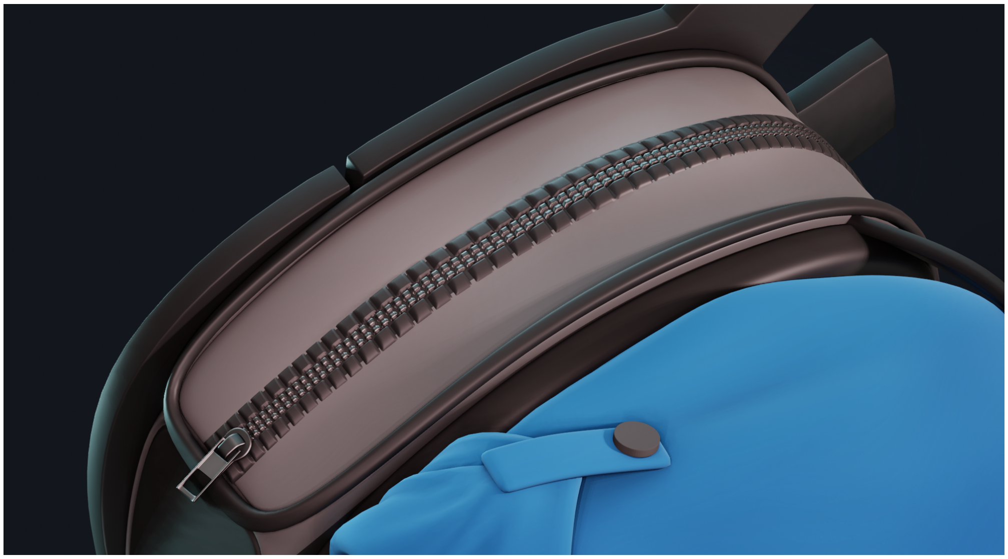
An example of using expand with mirror options, loops and a recursion to create wood carving patterns.

Tip

Mirror options can also be combined with the expansion.

Linear gradients `G` or brush falloff gradients `B` will help to add slanted surfaces to the patterns.

A "Recursions" with `R` or `Alt-R` will start a new expansion along the border of the current expansion. Doing this multiple times, can help for increasingly random patterns or advanced pattern creation.



An example of using loops and gradients with multiple expanded masks.

Tip

Remember that Expand only affects visible geometry. So if a pattern should only be created on a part of the mesh, hide the other geometry first.

Use the Mesh Filter to deform the geometry and the Color Filter to add colors, to apply the patterns on the sculpt.

Expanding Textures

Textures can be used to affect the shape and gradients of the expanded mask. This feature can be combined with loops and recursion to create unique results.

To use a texture, you need to assign it to your currently active brush in the Texture Brush Settings. The texture can be edited/created in the Texture Properties.

Note

This texture only works when the Mapping is set to *3D*.

Use `Y` and `T` to increase or decrease the affect the texture has on the edge of the mask.

Controls

Invert::	`F` Flips between expanding a positive mask (value of one) or a negative mask (value of zero). In the case of face sets, this option flips between including areas inside the masked area or areas outside the masked area. .. needs visual technical examples for both
Toggle Preserve State::	

	`E` Accumulate the new mask on top of the previous one or replace it. For Face Sets, this will toggle between creating Face Sets boundaries or replacing the existing Face Sets.
Move Origin::	`Spacebar` Moves the initial vertex used for calculating the falloff. .. needs visual technical example
Geodesic Falloff::	`1` Uses a falloff based on the Geodesic distances from the edge boundary to the active vertex.
Topology Falloff::	`2` Uses a falloff based on a flood fill using edges.
Diagonals Falloff::	`3` Uses a falloff based on a flood fill using polygon diagonals and edges.
Spherical Falloff::	`4` Uses a falloff based on the Euclidean distances from the edge boundary to the active vertex. .. needs visual technical examples
Toggle Gradient::	`G` Enables linear gradient of values from the origin to the current active vertex.
Toggle Brush Gradient::	`B` Similar to linear gradient but uses the current brush Falloff to define the shape of the falloff. .. needs visual technical examples
Geodesic Recursive Step::	`R` Start a new expansion with a Geodesic falloff from the boundary of the current falloff.
Topology Recursive Step::	`Alt-R` Start a new expansion with a topology falloff from the boundary of the current falloff. .. needs visual technical examples
Snap Expanded to Face Sets::	`Ctrl` Isolates the expanded region to the boundary of the face set under the cursor.
Loop Count Increase::	`W` Increase the number of loops or iterations the operator is run; using four loops will split the mask into four parts.
Loop Count Decrease::	

`Q` Decrease the number of loops or iterations the operator is run; using four loops will split the mask into four parts. .. needs visual technical examples

Texture Distortion Increase::

`Y` Increases the falloff distance when using a texture to distort the mask shape.

Texture Distortion Decrease::

`T` Decreases the falloff distance when using a texture to distort the mask shape. ..needs visual technical examples

Expand Mask by Normals

Reference

Mode::	Sculpt Mode
Menu::	*Mask ‣ Expand Mask by Normals*
Shortcut::	`Shift-Alt-A`

Expand a mask, following the curvature of the surface. This operator uses the same internal operator as Expand meaning all the shortcuts and functionality works the same as that tool.

This operator is especially useful for hard surface sculpting.

Tip

If one expansion does not properly fill the entire desired surface, use the operator repeatedly with a different starting point.

Note

Using any of the Falloff shortcuts `1–4` will replace the curvature falloff of this operator.

Expand Face Set by Topology

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Expand Face Set by Topology*
Shortcut::	`Shift-W`

Expands a face set from the active vertex. This operator uses the same internal operator as Expand meaning all the hotkeys and functionality works the same as that tool, with the gradient features as the exception.

Usage

Saving Masks

Expanding Face Sets has all the same use cases as expanding masks. The advantage for this one is that they will be saved for repeated usage. Face sets can be filled any time with a mask, so assigning areas face sets will save you time. (And of course face sets can be used to hide face sets)

Pivot Points for Pose Brush

When using the Pose Brush it is most predictable when using it with Face Sets to define the face set boundaries as pivot point locations. Face Sets can be expanded from a point or from a boundary between hidden face sets to create them quickly. Alternatively Grow/Shrink Face Sets or use the Expand Active Face Set to dynamically grow/shrink them.

Cloth Sculpting

Tools like the Cloth Filter and Cloth Brush work especially well when only simulating small areas at a time. Face Sets can very easily be created with Expand to assign areas of action.

Expand Active Face Set

Reference

Mode::	Sculpt Mode
Menu::	*Face Sets ‣ Expand Face Set by Topology*
Shortcut::	`Shift-Alt-W`

Expands an existing face set with a geodesic falloff. This operator uses the same internal operator as Expand meaning all the hotkeys and functionality works the same as that tool.

Note

Using any of the Falloff shortcuts `1-4` the operator to switch to Expand Face Set by Topology.

Usage

Resizing Face Sets

Resize a Face Set along the surface distances. It is an alternative to Grow/Shrink Face Sets which follows the topology instead of geodesic distances.

Joining Face Sets

When holding `Ctrl` the expansion will instead snap to other Face Sets. This is a fast way of joining multiple face sets into one.

Texture Paint

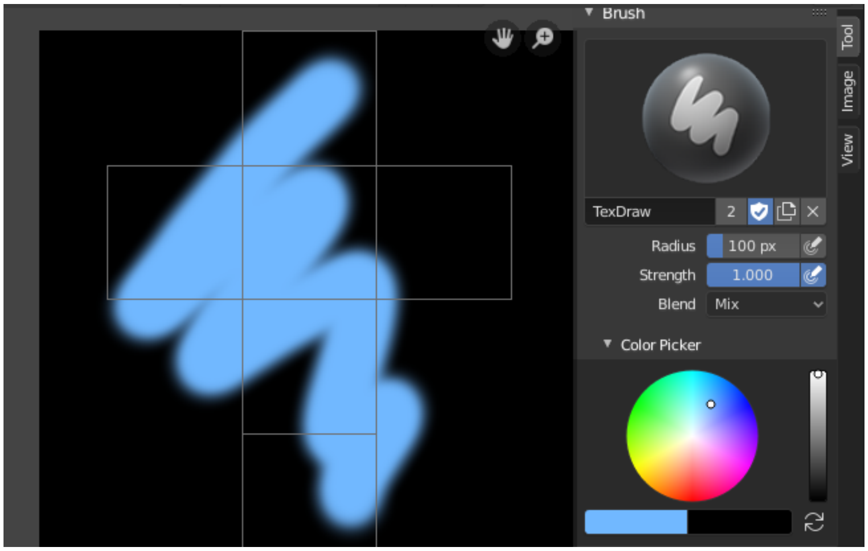
- Introduction
    - Getting Started
    - Texture Preview
    - Saving
    - Using an External Image Editor
    - Known Limitations
- Brushes
    - Brush Types
- Tool Settings
    - Texture Slots
    - Brushes
    - Brush Settings
    - Mask
    - Symmetry
    - Options
    - Tiling

Introduction

A UV texture is a picture (image, sequence or movie) that is used to color the surface of a mesh. The UV texture is mapped to the mesh through one or more UV maps. There are three ways to establish the image used by the UV texture:

1. Use any image editing program to create an image. In the Image Editor, select the UV texture and load the image. Blender will then use that texture's UV map to transfer the colors to the faces of the mesh.
2. Paint a flat image in the Image Editor onto the currently selected UV texture, using its UV map to transfer the colors to the faces of the mesh.
3. Paint the mesh in the 3D Viewport, and let Blender use the currently selected UV map to update the UV texture (as discussed below).

Blender features a built-in paint mode called *Texture Paint* which is designed specifically to help you edit your UV textures and images quickly and easily in either the Image Editor or the 3D Viewport. Since a UV texture is just a special-purpose image, you can also use any external paint program, like GIMP or Krita.



TexDraw 2

Radius 100 px

Strength 1.000

Blend Mix

Since a mesh can have layers of UV textures, there may be many images that color the mesh. However, each UV texture only has one image.

Texture Paint works in both a 3D Viewport and the

Image Editor. In the 3D Viewport in Texture Paint Mode, you paint directly on the mesh by projecting onto the UVs.

Texture painting in Blender.

Tip

Memory Optimization

Texture Paint is fast and responsive when working in the 3D Viewport and when your image is sized as a square where the side lengths are a power of two, e.g. 256×256, 512×512, 1024×1024, etc.

Getting Started

The object to be painted on must first be unwrapped. UVs can be added traditionally, with standard Unwrapping Tools, or by adding Simple UVs in Texture Paint mode.

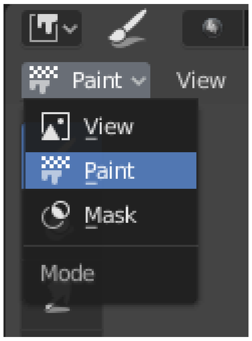
Note

When no UV layers can be detected, Blender will display a warning message.

Once you have unwrapped your model to a UV map, you can begin the texturing process. To use texture paint you may do any of the following:

- Activate the Texture Paint workspace. Here the 3D Viewport has the Texture Paint Mode enabled and the Image Editor is already switched to Paint mode.
- In the 3D Viewport, select Texture Paint Mode from the mode selector in the header, and you can paint directly onto the mesh.
- In the Image Editor, switch the mode to Paint (shown in the image to the right).

Once you enable Texture Painting, your mouse becomes a brush. As soon as you enable Texture Painting or switch to Texture Paint Mode, different tools become available in the Toolbar.



Enabling Paint mode.

In the Image Editor, you paint on a flat canvas that is wrapped around the mesh using UV coordinates. Any changes made in the Image Editor show up immediately in the 3D Viewport, and vice versa.

To work with the UV layout (for example, to move coordinates) you must use the UV Editor.

A full complement of brushes and colors can be selected from the Sidebar region in the Image Editor. Brush changes made in either panel are immediately reflected in the other panel. However, the modified texture will **not** be saved automatically; you must explicitly do so with Save Image.

Texture Preview

If your texture is already used to color, bump map, displace, alpha-transparent, etc., a surface of a model in your scene (in other technical words, is mapped to some aspect of a texture via a texture channel using UV as a map input), you can see the effects of your painting in the context of your scene as you paint.

To do this, set up side-by-side areas, one Area in 3D Viewport set to *Texture* shading option, and in the second Area the Image Editor loaded with your image. Position the 3D Viewport to show the object that is UV-mapped to the loaded image. In the image to the right, the texture being painted is mapped to the “Normal” attribute, and is called “bump mapping”, where the grayscale image is used to make the flat surface appear bumpy. See Texture Mapping Output for more information on bump mapping.

Saving

If the header menu item Image has an asterisk next to it means that the image has been changed, but not saved. Use Save Image or Save Image As to save your work with a different name or overwrite the original image.

Note

UV Textures

Since images used as UV textures are functionally different from other images, you should keep them in a directory separate from other images.

The image format for saving is independent of the format for rendering. The format for saving a UV image is selected in the header of the File Browser, and defaults to `PNG` (`.png`).

If Packing is enabled in the File Browser's header, or if you manually pack, saving your images to a separate file is not necessary.

Using an External Image Editor

If you use an external program to edit your UV texture, you must:

1. Run that paint program (GIMP, Krita, etc.).
2. Load the image or create a new one.
3. Change the image.
4. And re-save it within that program.
5. Back in Blender, you reload the image in the Image Editor.

You want to use an external program if you have teams of people using different programs that are developing the UV textures, or if you want to apply any special effects that Texture Paint does not feature, or if you are much more familiar with your favorite paint program.

Known Limitations

UV Overlap

In general overlapping UVs are not supported (as with texture baking).

However, this is only a problem when a single brush stroke paints onto multiple faces that share a texture.

Perspective View & Faces Behind the View

When painting onto a face which is partially behind the view (in perspective mode), the face cannot be painted on. To avoid this, zoom out or use an orthographic viewport.

Perspective View & Low Poly

When painting onto a face in perspective mode onto a low-poly object with normals pointing away from the view, painting may fail; to workaround disable the Normal Falloff option in the stroke settings.

Typically this happens when painting onto the side of a cube (see Blender bug #34665).

Brushes

Brush Types

See Brush Type.

Available brush types are listed here, together with brushes from the *Essentials* asset library using them.

Draw

  Brushes: Paint Hard, Paint Soft, Paint Hard Pressure, Paint Soft Pressure, Erase Hard, Erase Soft, Erase Hard Pressure

  The normal brush, draws a swath of color.

Soften

  Brushes: Blur

  Uses a “blur effect” to soften or sharpen the image.

  Direction

    Soften

      Is used to paint a blur effect.

      Kernel Radius (2D only)

        Blur radius in pixels.

    Sharpen

      The Sharpen tool enhances the contrast of the image as you paint over it.

      Sharp Threshold

        The Threshold will only apply sharpening to only those pixels that differ more than the threshold value from their surrounding pixels.

Kernel Radius (2D only)

The kernel size controls how big an area the tool searches over is while calculating that difference.

Blur Mode

The blur kernel type controls how neighboring pixels are weighted when calculating the blur effect.

Gaussian::	Gaussian will sample the pixels near the center of the brush most.
Box::	Box samples all surrounding pixels equally.

Smear

Brushes: Smear

When you click, takes the colors under the cursor, and blends them in the direction you move the mouse. Similar to the “smudge” tool of *Gimp*.

Clone

Brushes: Clone

Copies the colors from the specified image (or location of the same image) to the active image.

In 3D projective painting the clone cursor can be set with `Ctrl-LMB`. In 2D painting the clone can be moved dragging it with `RMB`.

Clone from Paint Slot (3D projective only)

Use another image as clone source, instead of using the 3D cursor position as the source in the same image.

Source Clone Slot

This allows you to select an image as a clone source.

Image (2D only)

Image used as a clone source.

Alpha (2D only)

Opacity of the clone image display.

Fill

Brushes: Fill

It can be used to fill large areas of the image with the brush color. The tool fills adjacent pixels that have a color value similar to the pixel you clicked on.

Fill Threshold (2D only)

Determines how much the color must be similar to the color of pixel you click to be filled. A low *Threshold* only fills very similar in color pixels. A higher *Threshold* fills pixels within a broader range of color.

The *Gradient* type of the Color Picker allows the use of a gradient to fill the image.

To apply the gradient with the *Fill* brush click `LMB` and drag to define the gradient line, or radius if a radial gradient is used (depending on the *Gradient Fill Mode*).

Gradient Fill Mode

Linear, Radial

Note

Overrides

For projective texturing it will bypass some options for projective painting to paint the model. This means that occluded, backfacing and normal culled faces will always get filled, regardless of whether the options are activated in the External panel.

Mask

Brushes: Mask

This brush paints gray-scale values on the mask texture specified in the Mask panel. Any masked surfaces will not be affected by other paint brushes, similar to sculpt mode masking.

Mask Value
: Mask weight, a value of zero means not masked, while one is completely masked. Hold `Ctrl` to invert the painted mask value.

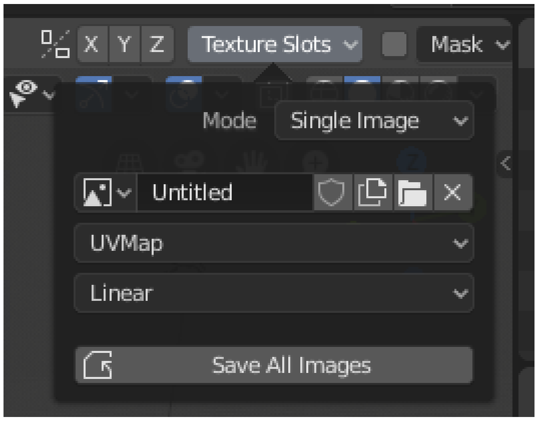
Tip

A simpler alternative is to use the face selection mask. See Face Selection Masking for details.

Tool Settings

- Texture Slots
- Brushes
  - Introduction
  - Manage Brushes
  - Brush Settings
- Brush Settings
- Mask
  - Stencil Mask
  - Cavity Mask
- Symmetry
- Options
  - External
- Tiling

Texture Slots



Mode: Single Image

Untitled

UVMap

Linear

Texture Slots settings.

The combination of images associated with UV maps is called “slots”.

Selecting a *Paint Slots* or *Canvas Image* will also display the corresponding image in the Image Editor.

Mode

The slot system includes two painting modes:

Material

This mode tries to detect the slots from the materials of the mesh.

For the Cycles renderer, all textures (*Image Texture* node) in the material’s node tree are added in the slots tab.

Active Paint Texture Index

A List view of slots. Activate a certain slot to use it for painting by `LMB` click on it.

Single Image

You can just select an existing image and painting will use the active UV layer for painting.

Image

Allows you to select the image used as a canvas.

New

Create a new image.

UV Map

Allows you to select the UV layer for painting. (Same as the currently active UV map in the mesh’s *UV Maps* panel.)

Texture Filter Type

Set the interpolation mode of the texture. This can be Linear or Closest.

Save All Images

Repack (or save if external file) all edited images. Same as in the Image Editor.

Add Simple UVs

The *Add Simple UVs* does a simple cube unwrap followed by a pack operation. It’s still recommended to make a custom unwrap.

This operator is available when the object does not already have a UV Map.

Brush Settings

Information on brush settings for every mode can be found in these pages:

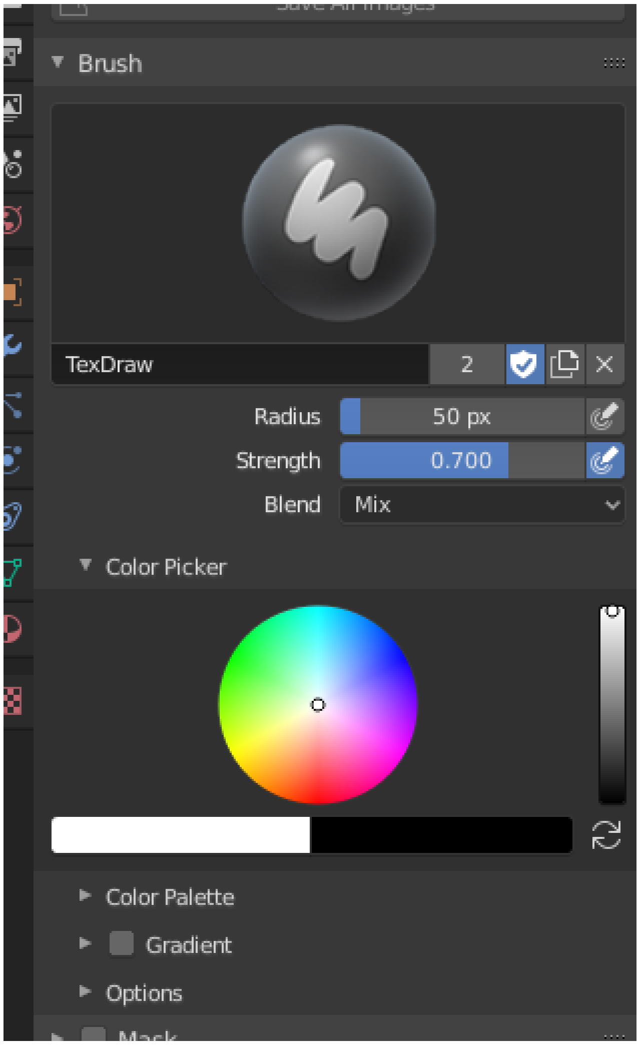
Brush
: General and advanced settings.

Texture
: Color and mask texture settings.

Stroke
: Stroke methods and settings.

Falloff
: Falloff curves and settings.

Cursor
: Cursor and appearance settings.


Brush
Color Picker
Color Palette
Gradient
Options

TexDraw 2

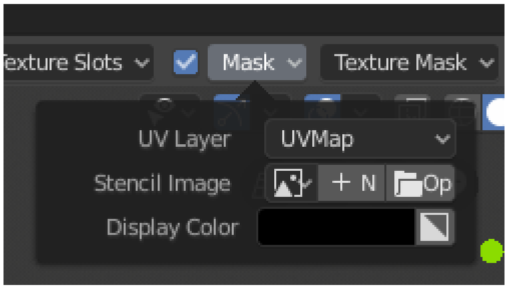
Radius 50 px

Strength 0.700

Blend Mix

Brush settings panel.

Mask


exture Slots
Mask
Texture Mask
Stencil Image
+ N
Op
Display Color

UV Layer: UVMap

Mask settings.

Stencil Mask

Specify an additional image texture that defines masked surfaces. Masked surfaces can be defined with the Mask brush and will not be affected by painting.

The mask can be deactivated by the checkbox in the header.

Stencil Image

  Image used as a mask. See Data-Block Menu.

UV Layer

  Allows you to select the UV layer for the mask image.

Display Color

  Mask color in the viewport. See Color Picker.

  Invert Stencil (black/white icon)

    Inverts the mask.

Cavity Mask

Cavity masking means that the brush will be masked if there is a cavity or a hill on the mesh surface depending on the mesh options. The cavity algorithm is vertex-based.

Symmetry

Reference

Mode::	Texture Paint Mode
Tool::	*Toolbar ‣ Tool ‣ Symmetry*

Mirror

Mirror the brush strokes across the selected local axes. Note that if you want to alter the directions the axes point in, you must rotate the model in Edit Mode and not in Object Mode.

Options

Bleed
: Seam Bleed extends the paint beyond UV island bounds to avoid visual artifacts (like bleed for baking).

Dither
: Amount of dithering when painting on 8-bit images.

Occlude
: With Geometry occlusion active only exposed (not hidden by other mesh parts) pixels are affected. This also allows for 3D stencils to be used to mask out areas of the surface too.

Backface Culling
: With backface culling enabled you can only paint on the front side of faces.

See also

See the Brush Display options.

External

Screen Grab Size
: Size of the captured image for reprojecting.

Quick Edit
: Edit a snapshot of the viewport in an external image editor.

Apply
: Project edited image back onto the object.

Apply Camera Image

Project an edited render from the active camera back onto the object.

Tiling

Reference

Editor::	Image Editor
Mode::	Paint Mode
Menu::	*Sidebar* ▸ *Tools* ▸ *Tiling*

Wraps the stroke to the other side of the image as your brush moves off the opposite side of the canvas. Very handy for making seamless textures.

X
left/right

Y
top/bottom

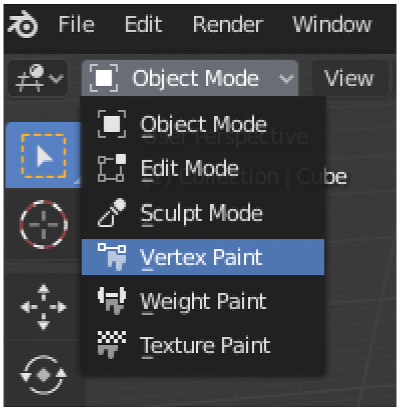
Vertex Paint

- Introduction
    - Viewing Color Attributes
- Brushes
    - Brush Types
- Vertex Paint Tools
- Tool Settings
    - Brushes
    - Brush Settings
    - Symmetry
- Editing
    - Smooth Vertex Colors
    - Dirty Vertex Colors
    - Vertex Color from Weight
    - Invert
    - Levels
    - Hue/Saturation/Value
    - Brightness/Contrast
    - Set Vertex Colors
    - Sample Color

Introduction

Vertex Painting is a simple way of painting color onto an object, by directly manipulating the color of vertices, rather than textures, and is fairly straightforward. Vertex Painting stores the color information as a Color Attribute which can be used by different render engines.

Color attribute’s can be managed in the pallette pop-over in the middle of the header.


File
Edit
Render
Window
Object Mode
View
Object Mode
Edit Mode
Sculpt Mode
Vertex Paint
Weight Paint
Texture Paint

Vertex Painting Mode.

When a vertex is painted, the color of the vertex is modified according to the settings of the brush. The color of all visible planes and edges attached to the vertex are then modified with a gradient to the color of the other connected vertices. Note that the color of occluded faces is not modified.

See also

Dynamic Paint can create Color Attribute information while using physics or animation.

Viewing Color Attributes

Color Attributes can be used in a material node tree using the Color Attribute Node.

Color Attributes can be viewed in the 3D viewport using the Workbench render engine. To use such feature, set the 3D Viewport to Solid Shading and select the *Attribute* Color option.

Brushes

Brush Types

See Brush Type.

Available brush types are listed here, together with brushes from the *Essentials* asset library using them.

Paint Vertex

> Brushes: Paint Hard, Paint Soft, Paint Hard Pressure, Paint Soft Pressure, Airbrush
>
> Paints a specified color over the object.

Blur

> Brushes: Blur
>
> Smooths out the colors of adjacent vertices. In this mode the Color Value is ignored. The strength defines how much the colors are blurred.

Average

> Brushes: Average
>
> Smooths color by painting the average resulting color from all colors under the brush.

Smear

> Brushes: Smear
>
> Smudges colors by grabbing the colors under the brush and “dragging” them. This can be imagined as a finger painting tool.

Vertex Paint Tools

Brush

Tool to use for any of the vertex paint brushes.

Annotate

Draw free-hand annotation.

Annotate Line

Draw straight line annotation.

Annotate Polygon

Draw a polygon annotation.

Annotate Eraser

Erase previous drawn annotations.

Tool Settings

- Brushes
    - Introduction
    - Manage Brushes
    - Brush Settings
- Brush Settings
- Symmetry

Brush Settings

Information on brush settings for every mode can be found in these pages:

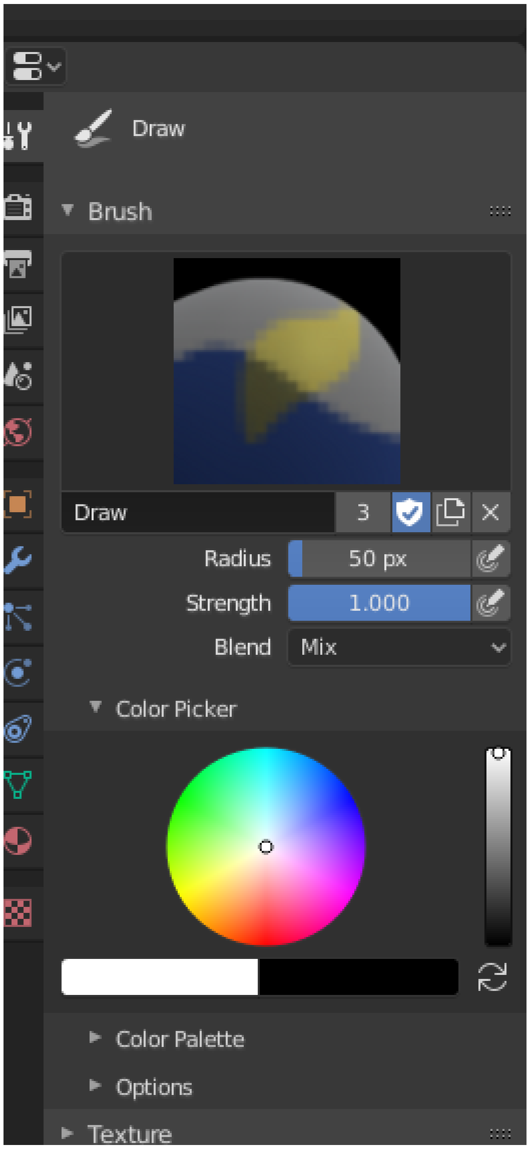
Brush
: General and advanced settings.

Texture
: Color and mask texture settings.

Stroke
: Stroke methods and settings.

Falloff
: Falloff curves and settings.

Cursor
: Cursor and appearance settings.


Draw
Brush
Draw
3
Color Picker
Color Palette
Options
Texture

Radius 50 px
Strength 1.000
Blend Mix

Brush settings panel.

Symmetry

Reference

Mode::	Vertex Paint Mode
Tool::	*Toolbar* ‣ *Tool* ‣ *Symmetry*

Mirror

Mirror the brush strokes across the selected local axes. Note that if you want to alter the directions the axes point in, you must rotate the model in Edit Mode and not in Object Mode.

Editing

Smooth Vertex Colors

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Smooth Vertex Colors*

Smooth colors across vertices.

Dirty Vertex Colors

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Dirty Vertex Colors*

Generate a dirt map gradient based on cavity.

Blur Strength
: Blur strength per iteration.

Blur Iterations
: Number of times to blur the colors (higher blurs more).

Highlight Angle
: Clamps the angle for convex areas of the mesh. Lower values increase the contrast but can result in clamping. 90 means flat, 180 means infinitely pointed.

Dirt Angle

Clamps the angle for concave areas of the mesh. Higher values increase the contrast but can result in clamping. 90 means flat, 0 means infinitely deep.

Dirt Only

When active it won't calculate cleans for convex areas.

Normalize

Choose optimal contrast by effectively lowering *Highlight Angle* and increasing *Dirt Angle* automatically. Disabling *Normalize* allows getting consistent results across multiple objects.

Vertex Color from Weight

Reference

Mode:: Vertex Paint Mode
Menu:: *Paint ‣ Vertex Color from Weight*

Converts the active weight into grayscale colors.

Invert

Reference

Mode:: Vertex Paint Mode
Menu:: *Paint ‣ Invert*

Invert RGB values.

Levels

Reference

Mode:: Vertex Paint Mode

Menu::	*Paint ‣ Levels*

Adjust the levels of the selected vertices.

Hue/Saturation/Value

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Hue/Saturation/Value*

Adjust the HSV values of the selected vertices.

Brightness/Contrast

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Brightness/Contrast*

Adjust the brightness/contrast of the selected vertices.

Set Vertex Colors

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Set Vertex Colors*
Shortcut::	`Ctrl-X`

Fill the active Color Attribute with the current paint color.

Affect Alpha
: Set color completely opaque instead of reusing existing alpha.

Sample Color

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Sample Color*
Shortcut::	`Shift-X`

Adjust the brush color of the Draw tool to the color under the mouse cursor.

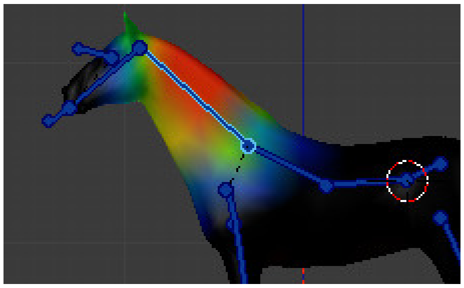
Weight Paint

- Introduction
  - The Weighting Color Code
  - Normalized Weight Workflow
- Brushes
  - Brush Types
- Weight Paint Tools
- Tool Settings
  - Brushes
  - Brush Settings
  - Symmetry
  - Options
- Using Vertex Groups
  - Vertex Groups for Bones
  - Vertex Groups for Particles
- Editing
  - Assign from Bone Envelopes
  - Assign Automatic from Bone
  - Normalize All
  - Normalize
  - Mirror
  - Invert
  - Clean
  - Quantize
  - Levels
  - Smooth
  - Transfer Weights
  - Limit Total
  - Set Weight
  - Sample Weight
  - Sample Group
  - Gradient (Linear)
  - Gradient (Radial)
  - Locks

Introduction

Vertex Groups can potentially have a very large number of associated vertices and thus a large number of weights (one weight per assigned vertex). *Weight Painting* is a method to maintain large amounts of weight information in a very intuitive way.

It is primarily used for rigging meshes, where the vertex groups are used to define the relative bone influences on the mesh. But we use it also for controlling particle emission, hair density, many modifiers, shape keys, etc.



Vertex group in Weight Paint Mode.

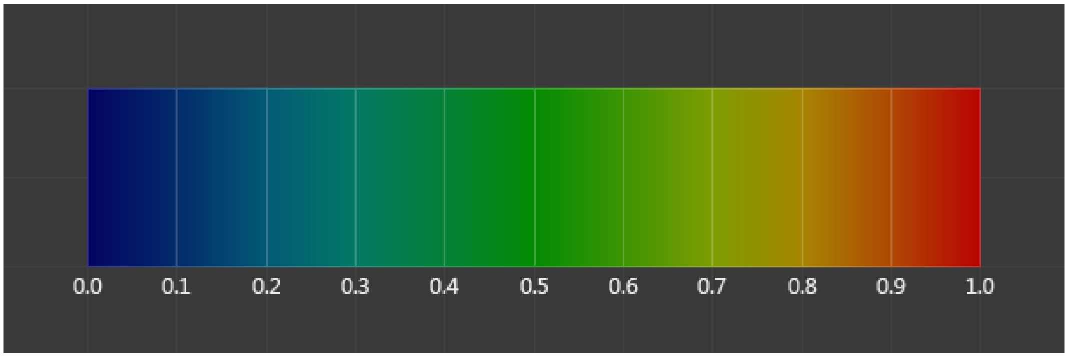
You can enter Weight Paint Mode from the Mode selector `Ctrl-Tab`. The selected mesh object is displayed slightly shaded with a rainbow color spectrum. The color visualizes the weights associated to each vertex in the active vertex group. By default *blue* means unweighted and *red* means fully weighted.

You can assign weights to the vertices of the object by painting on it with weight brushes. Starting to paint on a mesh automatically adds weights to the active vertex group (a new vertex group is created if needed).

Vertex Groups can be managed in the pallette pop-over in the middle of the header.

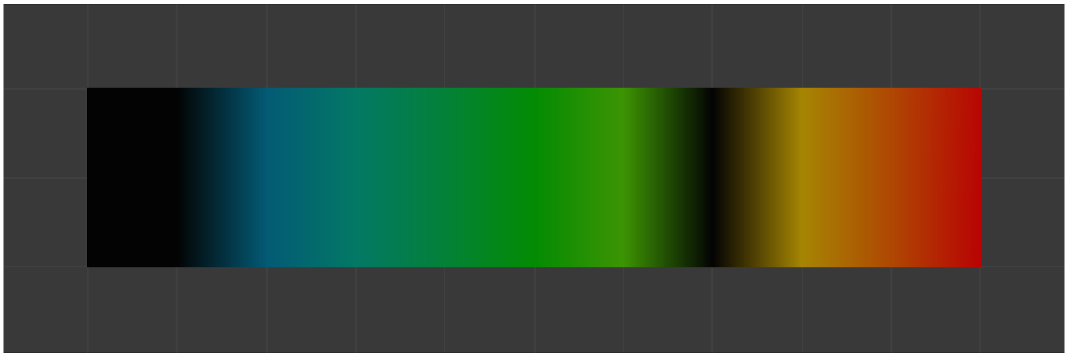
The Weighting Color Code

Weights are visualized by a gradient using a cold/hot color system, such that areas of low value (with weights close to 0.0) are displayed as blue (cold) and areas of high value (with weights close to 1.0) are displayed as red (hot). And all in-between values are displayed as rainbow colors (blue, green, yellow, orange, red).



The color spectrum and their respective weights.

In addition to the above described color code, Blender has a special visual notation (as an option) for unreferenced vertices: They are displayed as black. Thus you can see the referenced areas (displayed as cold/hot colors) and the unreferenced areas (in black) at the same time. This is most practicable when you look for weighting errors. See Options.



Unreferenced vertices example.

Note

You can customize the colors in the weight gradient by enabling Custom Weight Paint Range in the *Editing* tab of the *Preferences*.

Normalized Weight Workflow

In order to be used for things like deformation, weights usually have to be normalized, so that all deforming weights assigned to a single vertex add up to 1. The *Armature* modifier in Blender does this automatically, so it is technically not necessary to ensure that weights are normalized at the painting stage.

However, while more complicated, working with normalized weights has certain advantages, because it allows use of certain tools designed for them, and because when weights are normalized, understanding the final influence of the current group does not require knowing weights in other groups on the same vertex.

These tools are provided to aid working with normalized weights:

Normalize All

In order to start working with normalized weights it is first necessary to normalize the existing weights. The Normalize All tool can be used for that. Make sure to select the right mode and disable *Lock Active*.

Auto Normalize

Once the weights are initially normalized, the Auto Normalize option can be enabled to automatically maintain normalization as you paint. This also tells certain tools that the weights are supposed to be already normalized.

Vertex group locking

Any vertex group can be locked to prevent changes to it. This can be done via the lock icon in the vertex group list, or using bone selection and the locks pie menu.

This setting prevents accidental edits to groups. However, since it is also respected by *Auto Normalize*, in the normalized weight workflow it has a more significant meaning of locking the current influence of chosen bones, so that when you paint other bones, the weight is redistributed only between the unlocked groups.

In locations affected by multiple bones, this allows more precise tweaking and re-balancing of weights by temporarily focusing on a subset of bones. This can also be aided by the Lock Relative option, which displays unlocked groups as though re-normalized with the locked groups deleted, thus making it appear as if the locked groups did not even exist.

Multi-Paint

Finally, the Multi-Paint option allows treating multiple selected bones as if they were one bone, so that the painting operations change the combined weight, preserving the ratio within the group. Combined with locking, this allows balancing between one set of bones versus the rest, excluding a third set that has its influence not affected in any way due to locks.

Technically, this option does not require the normalized workflow, but since non-normalized weights can add to more than 1, the weight display behaves best with *Auto Normalize* enabled.

Tip

For example, when dealing with a bone loop, e.g. mouth or an eye, selecting the loop with *Multi-Paint* exposes the falloff between the loop as a whole and surrounding bones, while locking the surrounding bones and using *Lock Relative* displays the falloff between bones within the loop. Thus the complex two-dimensional falloff of each bone can be viewed and edited as two independent one-dimensional gradients.

Brushes

Brush Types

See Brush Type.

Available brush types are listed here, together with brushes from the *Essentials* asset library using them.

Draw

Brushes: Paint

Paints a specified weight over the object.

Blend

The brush Blend Modes defines in which way the weight value is applied to the vertex group while painting.

Mix

In this Blending mode the Weight value defines the *target weight* that will eventually be reached when you paint long enough on the same location of the mesh. And the strength determines how many strokes you need to place at the target weight. Note that for strength = 1.0 the target weight is painted immediately and for Weight = 0.0 the brush just does nothing.

Add

In this Blending mode the specified weight value is *added* to the vertex weights. The strength determines which fraction of the weight gets added per stroke. However, the brush will not paint weight values above 1.0.

Subtract

In this Blending mode the specified weight value is *subtracted* from the vertex weights. The strength determines which fraction of the weight gets removed per stroke. However, the brush will not paint weight values below 0.0.

Lighten

In this Blending mode the specified weight value is interpreted as the target weight. Very similar to the Mix Blending mode, but only weights below the target weight are affected. Weights above the target weight remain unchanged.

Darken

This Blending mode is very similar to the Lighten Blending mode. But only weights above the target weight are affected. Weights below the target weight remain unchanged.

Multiply

Multiplies the vertex weights with the specified weight value. This is somewhat like subtract, but the amount of removed weight is now dependent on the Weight value itself.

Blur

Smooths out the weighting of adjacent vertices. In this mode the Weight Value is ignored. The strength defines how much the smoothing is applied.

Blur

Brushes: Blur

Smooths out the weighting of adjacent vertices. In this mode the Weight Value is ignored. The strength defines how much the smoothing is applied.

Average

Brushes: Average

Smooths weights by painting the average resulting weight from all weights under the brush.

Smear

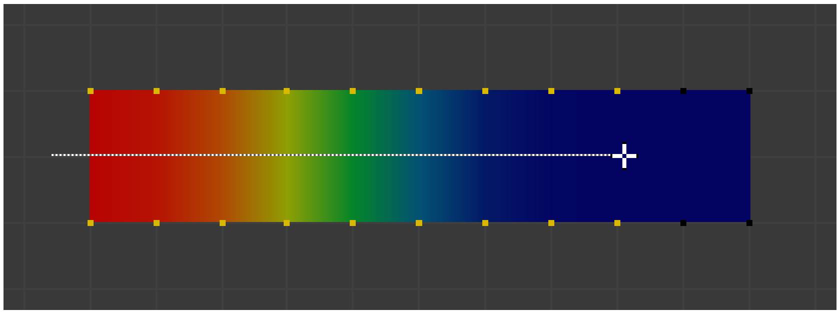
Brushes: Smear

Smudges weights by grabbing the weights under the brush and “dragging” them. This can be imagined as a finger painting tool.

Weight Paint Tools

Brush
: Tool to use for any of the weight paint brushes.

Gradient
: Applies a linear/radial weight gradient; this is useful at times when painting gradual changes in weight becomes difficult. Blends the weights of selected vertices with unselected vertices.



Example of the Gradient tool being used with selected vertices.

Weight
: The gradient starts at the current selected weight value, blending out to nothing.

Strength
: Lower values can be used so the gradient mixes in with the existing weights (just like with the brush).

Type
: The shape of the gradient.

Linear::	Create gradient that forms a straight line.
Radial::	Create gradient that forms a circle.

Note

These are also availible via shortcuts as the menu operators.

Sample

Weights

Sets the brush *Weight* as the weight selected under the cursor. The sampled weight is displayed in the tool settings.

Vertex Group

Displays a list of possible vertex groups to select that are under the cursor.

Annotate

Draw free-hand annotation.

Annotate Line

Draw straight line annotation.

Annotate Polygon

Draw a polygon annotation.

Annotate Eraser

Erase previous drawn annotations.

Tool Settings

- Brushes
    - Introduction
    - Manage Brushes
    - Brush Settings
- Brush Settings
- Symmetry
- Options

Brush Settings

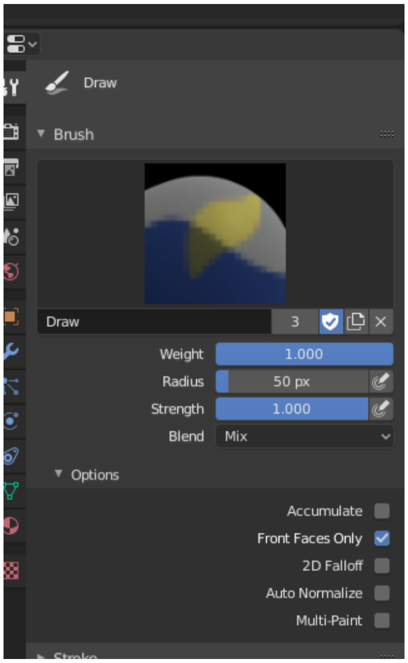
Information on brush settings for every mode can be found in these pages:

Brush
    General and advanced settings.

Stroke
    Stroke methods and settings.

Falloff
    Falloff curves and settings.

Cursor
    Cursor and appearance settings.



Weight 1.000
Radius 50 px
Strength 1.000
Blend Mix
Accumulate ☐
Front Faces Only ☑
2D Falloff ☐
Auto Normalize ☐
Multi-Paint ☐

Brush settings panel.

Symmetry

Reference

Mode:: Vertex Paint Mode
Tool:: *Toolbar ‣ Tool ‣ Symmetry*

Mirror Vertex Groups

Use this option for mirrored painting on groups that have symmetrical names, like with suffix ".R"/ ".L" or "_R" / "_L". If a group has no mirrored counterpart, it will paint symmetrically on the active group itself. You can read more about the naming convention in Editing Armatures: Naming conventions. The conventions for armatures/bones apply here as well.

Mirror X, Y, Z

Mirror the brush strokes across the selected local axes. Note that if you want to alter the directions the axes point in, you must rotate the model in Edit Mode and not in Object Mode.

Radial X, Y, Z

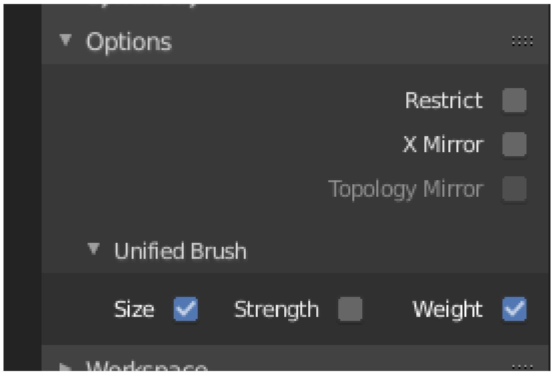
These settings allow for radial symmetry in the desired axes. The number determines how many times the stroke will be repeated within 360 degrees around the central axes.

Topology Mirror

Use topology-based mirroring, for when both sides of a mesh have matching mirrored topology. See here for more information.

Options

The weight paint options change the overall brush behavior.



Restrict ☐

X Mirror ☐

Topology Mirror ☐

Size ☑ Strength ☐ Weight ☑

Paint options.

Auto Normalize

Ensures that all deforming vertex groups add up to one while painting. When this option is turned off, then all weights of a vertex can have any value between 0 and 1. However, when vertex groups are used as deform groups for character animation then Blender always interprets the weight values relative to each other. That is, Blender always does a normalization over all deform bones. Hence in practice it is not necessary to maintain a strict normalization and further normalizing weights should not affect animation at all.

This option works most intuitively when used to maintain normalization while painting on top of weights that are already normalized with another tool.

Lock-Relative

Displays bone-deforming groups as if all locked deform groups were deleted, and the remaining ones were re-normalized. This is intended for use when balancing weights within a group of bones while all other bones are locked. With this option you can also temporarily view non-normalized weights as if they were normalized, without actually changing the values.

Multi-Paint

Paint on all selected vertex groups simultaneously, in a way that preserves their relative influence. This can be useful when tweaking weights in an area that is affected by more than three bones at once, e.g. certain areas on a character's face.

This option is only useful in the *Armature* tab, where you can select multiple vertex groups by selecting multiple pose bones. Once at least two vertex groups are selected, viewport colors and paint logic switch to Multi-Paint mode, using the sum of the selected groups' weights if *Auto Normalize* is enabled, and the average otherwise. Any paint operations aimed at this collective weight are applied to individual vertex group weights in such way that their ratio stays the same.

Since the ratio is undefined if all weights are zero, Multi-Paint cannot operate on vertices that do not have any weight assigned to the relevant vertex groups. For this reason it also does not allow reducing the weight all the way to zero. When used with *X Mirror*, it only guarantees completely a symmetrical result if weights are initially symmetrical.

Tip

While Multi-Paint cannot directly paint on zero-weight vertices, it is possible to use the *Smooth Weight* tool to copy a reasonable nonzero weight distribution from adjacent vertices without leaving Multi-Paint mode or changing bone selection.

To do that, enable vertex selection, select target vertices, and apply one iteration of the tool using vertex groups from *Selected Pose Bones* with low Factor. After that simply paint on top to set the desired collective weight.

Restript

This option limits the influence of painting to vertices (even with weight 0) belonging to the selected vertex group.

See also

See the Brush Display options.

Using Vertex Groups

Vertex Groups for Bones

This is one of the main uses of weight painting. While you can have Blender generate the weights automatically (see the skinning section), you may want to tweak them or even create them from scratch, especially around joints.

The process is as follows:

1. Select the armature and bring it into *Pose Mode* by pressing `Ctrl-Tab`.
2. Make sure that *Edit ‣ Lock Object Modes* is unchecked in the topbar.
3. Select the mesh and bring it into *Weight Paint Mode*.
4. Make sure that *Bone Selection* is checked in the 3D Viewport's header.
5. Select a bone using `Alt-LMB` (or `Shift-Ctrl-LMB`). This will activate the bone's vertex group and display its current weights on the mesh.
6. Paint weights for the bone using `LMB`.

Note

You can only select one bone at a time in this mode.

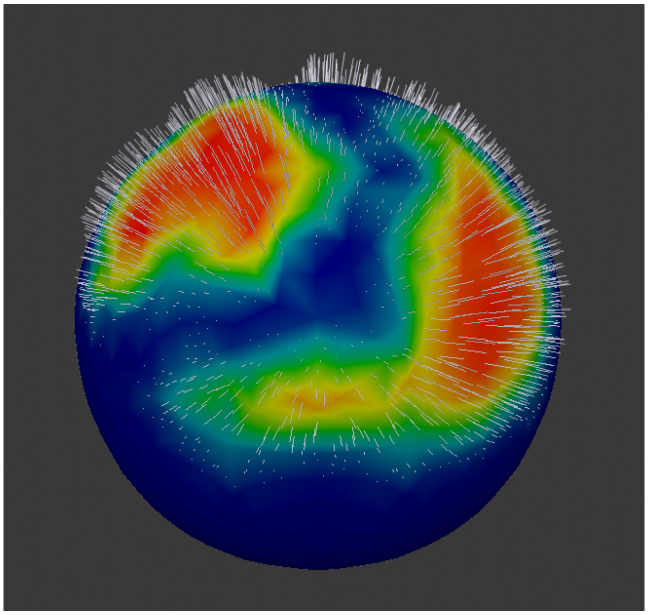
Tip

The bones are likely embedded inside the mesh, making them invisible and unselectable. To get around this, you can enable In Front for the armature.

If a bone doesn't have a vertex group yet when you start painting, Blender will create one automatically.

If you have a symmetrical mesh and a symmetrical armature, you can use Mirror Vertex Groups to automatically create vertex groups and weights for the other side.

Vertex Groups for Particles



Weight painted particle emission.

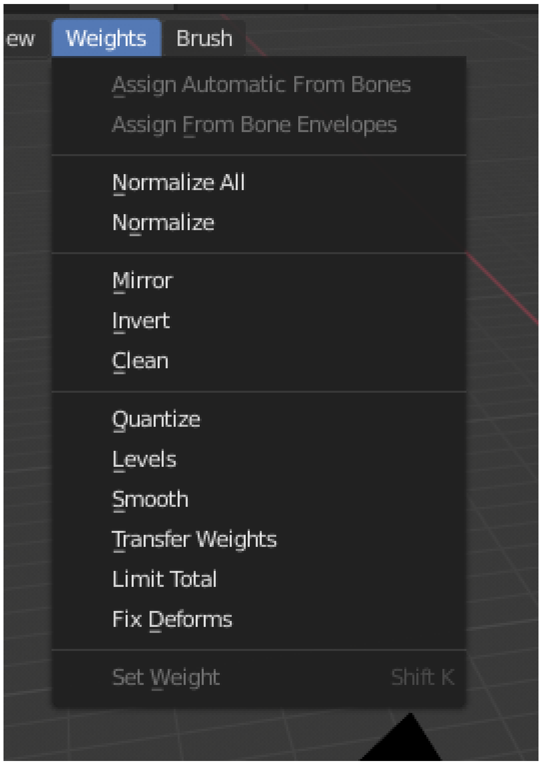
By selecting vertex groups in the Vertex Groups panel of a particle system's properties, you can have different particle densities, hair lengths etc. across different areas of the mesh.

Editing

Reference

Mode:: Edit Mode and Weight Paint Mode
Menu:: *Weights*

Blender provides a set of helper tools for Weight Painting.


ew
Weights
Brush
Assign Automatic From Bones
Assign From Bone Envelopes
Normalize All
Normalize
Mirror
Invert
Clean
Quantize
Levels
Smooth
Transfer Weights
Limit Total
Fix Deforms
Set Weight
Shift K

Weight Paint Tools.

The Subset Option

Some of the tools also provide a Subset filter to restrict their functionality to only specific vertex groups (in the Adjust Last Operation panel, displayed after the tool is called) with following options:

- Active Group
- Selected Pose Bones
- Deform Pose Bones
- All Groups

All tools also work with Vertex Selection Masking and Face Selection Masking. In these modes the tools operate only on selected vertices or faces.

Assign from Bone Envelopes

Apply the envelope weight of the selected bone(s) to the selected vertex group.

Assign Automatic from Bone

Apply from the selected bone(s) to the vertex group the same "auto-weighting" methods as available in the Parent armature menu.

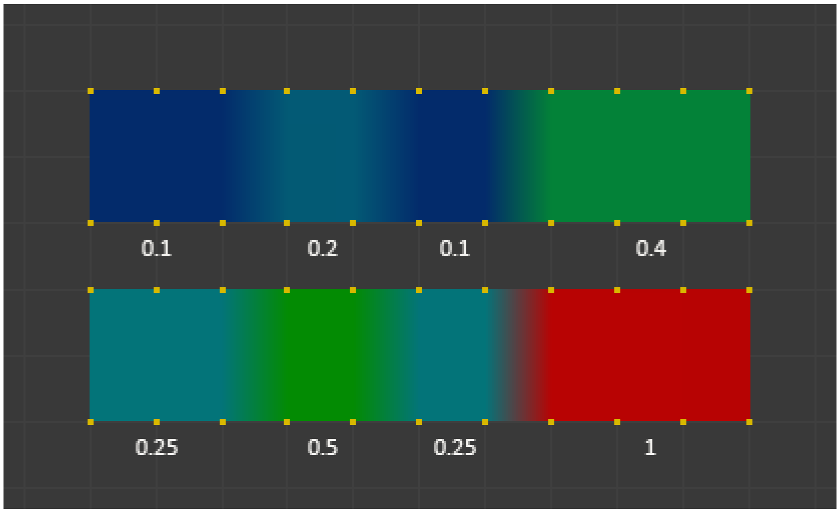
Normalize All

For each vertex, this tool makes sure that the sum of the weights across all vertex groups is equal to 1. This tool normalizes all of the vertex groups, except for locked groups, which keep their weight values untouched.

Lock Active
: Keep the values of the active group while normalizing all the others.

Normalize

This tool only works on the active vertex group. All vertices keep their relative weights, but the entire set of weights is scaled up such that the highest weight value is 1.0.



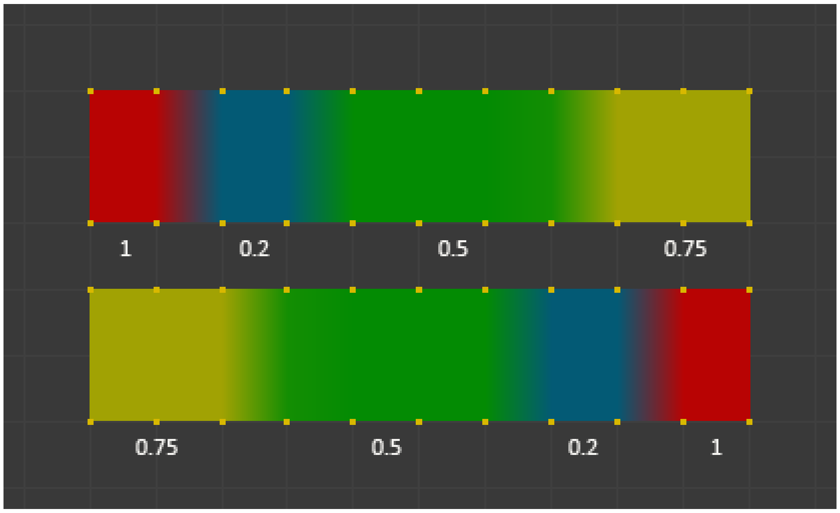
Normalize example.

Mirror

The *Mirror Vertex Group* tool mirrors the weights from one side of a perfectly symmetrical mesh to the opposite side. Those vertices that have no corresponding vertex on the other side will not be affected. But note, the weights are not transferred to the corresponding opposite bone weight group.

Note

Mirroring only works when the object's rest pose is perfectly symmetrical across the X axis.



Mirror example.

Mirror Weights

With this option checked, every selected vertex receives the weight information of its symmetrical counterpart. If both vertices are selected, it will be a weight information exchange; if only one is selected, information from the unselected will overwrite the selected one. Information on weight is passed for the active group only, unless *All Groups* is checked, in which case it is passed for all groups.

Flip Group Names

Works with selected vertices that belong to vertex groups with "symmetrical names" (with components like "L", "R", "right", "left").

All selected vertices that belong to the active group, or to the symmetrical of the active group, will have their assignation to that group replaced by an assignation to the symmetrical one; however, its weight will be preserved. If *All Groups* is checked, all assignations to these kind of groups will be replaced by the symmetrical counterpart, also keeping the old weights.

All Groups

Operate on all vertex groups, instead of the active one.

Topology Mirror

Mirror for meshes which are not fully symmetric (approximate mirror). See here for more information.

Tip

Mirror to Opposite Bone

If you want to create a mirrored weight group for the opposite bone (of a symmetric character), then you can do this:

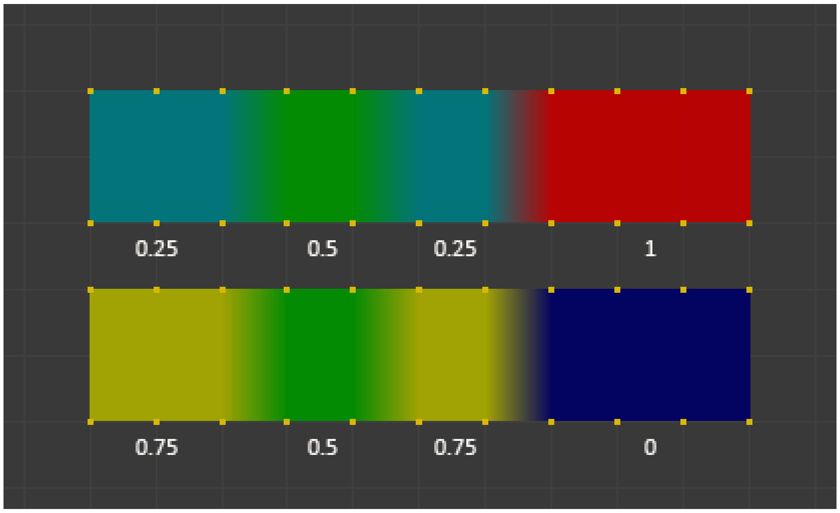
1. Delete the target vertex group (where the mirrored weights will be placed).
2. Create a copy of the source bone vertex group (the group containing the weights which you want to copy).
3. Rename the new vertex group to the name of the target vertex group (the group you deleted above).
4. Select the target vertex group and call the Mirror tool (use only the Mirror weights option and optionally *Topology Mirror* if your mesh is not symmetric).

Invert

Replaces each Weight of the selected weight group by × -1.0 weight.

Examples:

- Original 1.0 converts to 0.0
- Original 0.5 remains 0.5
- Original 0.0 converts to 1.0



Invert.

Subset
: Restrict the tool to a subset. See above The Subset Option about how subsets are defined.

Add Weights
: Add vertices that have no weight before inverting (these weights will all be set to 1.0).

Remove Weights
: Remove vertices from the vertex group if they are 0.0 after inverting.

Note

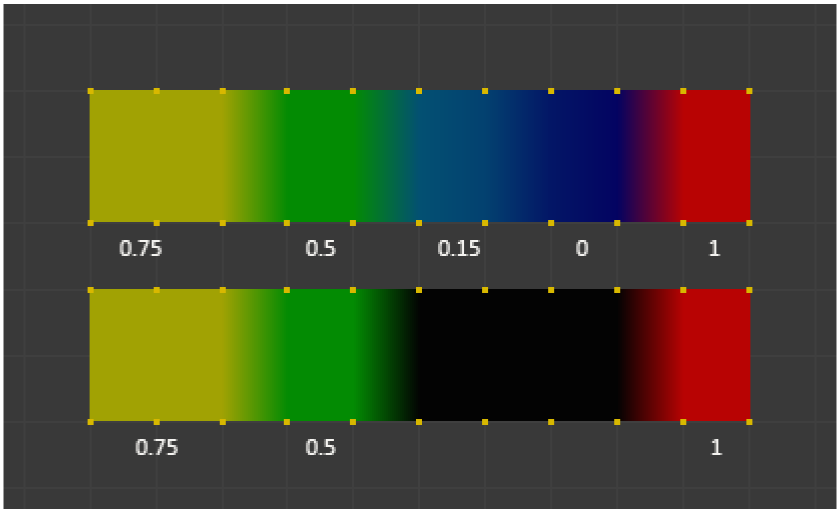
Locked vertex groups are not affected.

Clean

Clean Vertex Group Weights unassigns vertices from Vertex Groups whose weights are below the *Limit*. Removes weights below a given threshold. This tool is useful for clearing your weight groups of very low (or zero) weights.

In the example shown, a cutoff value of 0.2 is used (see operator options below) so all blue parts are cleaned out.

Note, the images use the *Show Zero weights* Active option so that unreferenced Weights are shown in Black.



Clean example.

Subset

Restrict the tool to a subset. See above The Subset Option for how subsets are defined.

Limit

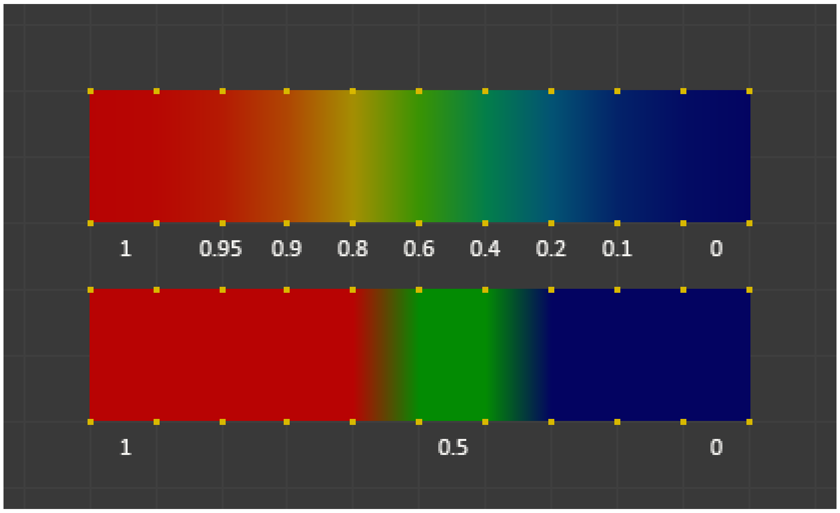
This is the minimum weight value that will be kept in the group. Weights below this value will be removed from the group.

Keep Single

Ensure that the *Clean* tool will not create completely unreferenced vertices (vertices which are not assigned to any vertex group), so each vertex will keep at least one weight, even if it is below the limit value!

Quantize

This operator uses a process known as Quantization which takes the input weights and clamps each weight to a number of steps between (0 - 1), so there is no longer a smooth gradient between values.



Quantize example (Steps = 2).

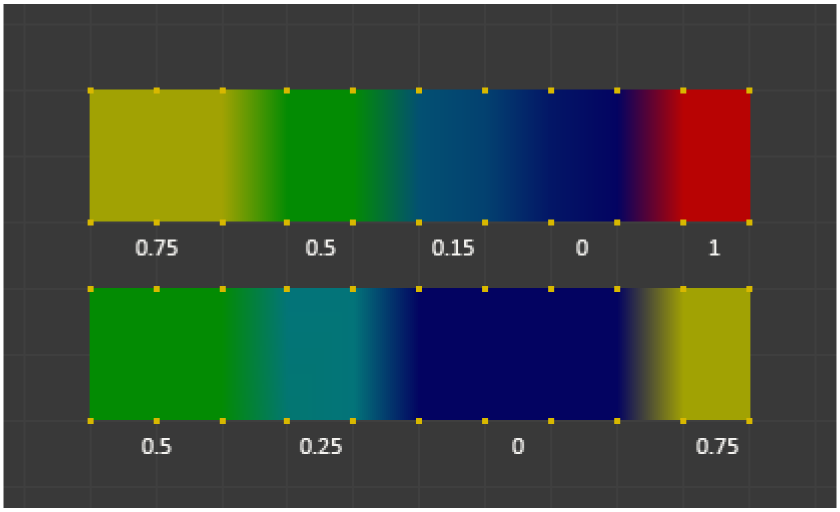
Steps
: The number of steps between 0 and 1 to quantize the weights into. For example 5 would allow the following weights `[0.0, 0.2, 0.4, 0.6, 0.8, 1.0]`.

Levels

Adds an offset and a scale to all weights of the selected weight groups. with this tool you can raise or lower the overall “heat” of the weight group.

Note

No weight will ever be set to values above 1.0 or below 0.0 regardless of the settings.



Levels example.

Subset

Restrict the tool to a subset. See above The Subset Option for how subsets are defined.

Offset

A value from the range (-1.0 - 1.0) to be added to all weights in the vertex group.

Gain

All weights in the Subset are multiplied with the gain.

Note

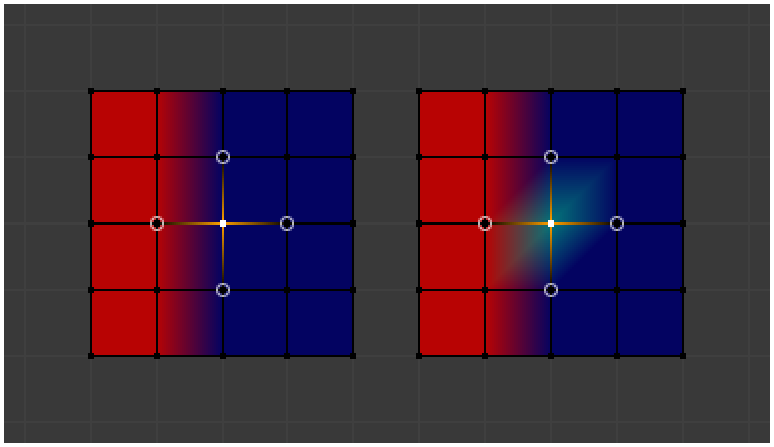
Whichever *Gain* and *Offset* you choose, in all cases the final value of each weight will be clamped to the range (0.0 - 1.0). So you will never get negative weights or overheated areas (weight > 1.0) with this tool.

Smooth

Tip

The Smooth tool only works when “Vertex selection masking for painting” is enabled. Otherwise the tool button is grayed out.

Blends the weights of selected vertices with adjacent unselected vertices. This tool only works in vertex select mode.

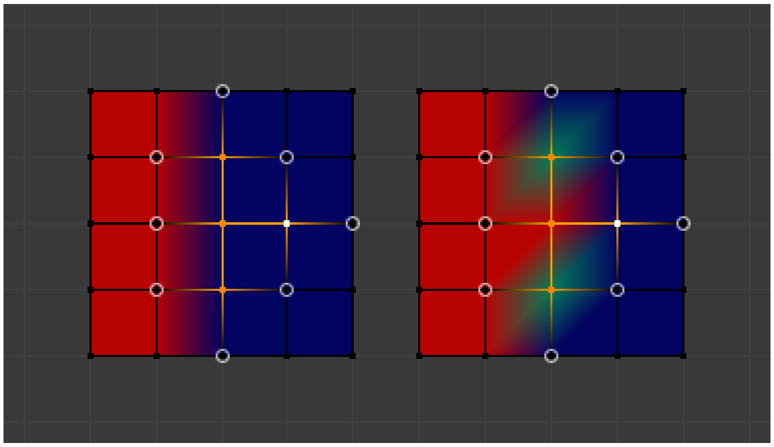


To understand what the tool really does, let us take a look at a simple example. The selected vertex is connected to four adjacent vertices (marked with a gray circle in the image). All adjacent vertices are unselected. Now the tool calculates the average weight of all connected **and** unselected vertices. In the example this is:

$(1 + 0 + 0 + 0) / 4 = 0.25$

This value is multiplied by the factor given in the Operator options (see below).

- If the factor is 0.0 then actually nothing happens at all and the vertex just keeps its value.
- If the factor is 1.0 then the calculated average weight is taken (0.25 here).
- Dragging the factor from 0 to 1 gradually changes from the old value to the calculated average.



Now let us see what happens when we select all but one of the neighbors of the selected vertex as well. Again all connected and unselected vertices are marked with a gray circle. When we call the Smooth tool now and set the Factor to 1.0, then we see different results for each of the selected vertices:

- The top-most and bottom-most selected vertices:

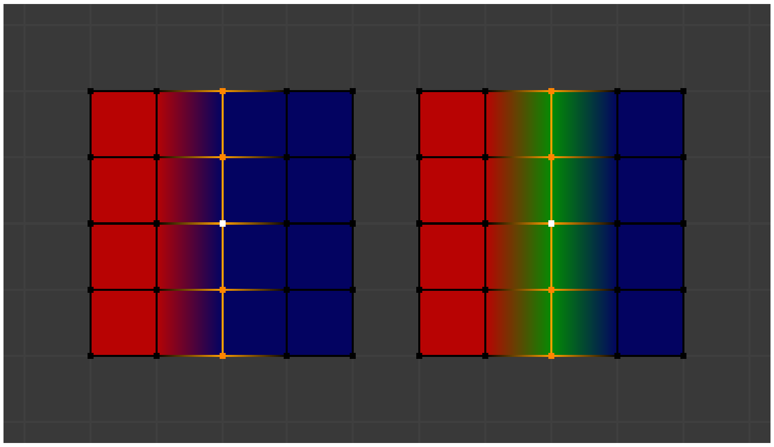
  are surrounded by three unselected vertices, with an average weight of \((1 + 0 + 0) / 3 = 0.333\) So their color has changed to light green.

- The middle vertex:

  is connected to one unselected vertex with `weight = 1`. So the average weight is 1.0 in this case, thus the selected vertex has changed to red.

- The right vertex:

  is surrounded by three unselected vertices with average weight = \((0 + 0 + 0) / 3 = 0.0\) So the average weight is 0, thus the selected vertex has not changed at all (it was already blue before Smooth was applied).



Finally let us look at a practical example. The middle edge loop has been selected and it will be used for blending the left side to the right side of the area.

- All selected vertices have two unselected adjacent vertices.
- The average weight of the unselected vertices is $(1 + 0) / 2 = 0.5$
- Thus when the *Factor* is set to 1.0 then the edge loop turns to green and finally does blend the cold side (right) to the hot side (left).

Factor

The effective amount of blending. When Factor is set to 0.0 then the Smooth tool does not do anything. For Factor > 0 the weights of the affected vertices gradually shift from their original value towards the average weight of all connected **and** unselected vertices (see examples above).

Iterations

Number of times to repeat the smoothing operation.

Expand/Contract

Positive values expand the selection to neighboring vertices while contract limits to the selection.

Source

The vertices to mix with.

All::	Smoothing will smooth both selected and deselected vertices.
Only Selected::	Smoothing will only smooth with selected vertices.
Only Deselected::	Smoothing will only smooth with deselected vertices.

Transfer Weights

Copy weights from other objects to the vertex groups of the active object.

By default this tool copies only the active (selected) vertex group of the source object to the active vertex group of target object or creates a new one if the group does not exist. However, you can change the tool’s behavior in the Adjust Last Operation panel.

For example, to transfer all existing vertex groups from the source objects to the target, change the *Source Layers Selection* option to *By Name*.

Note

This tool uses the generic “data transfer”, but transfers from all selected objects to active one. Please refer to the Data Transfer docs for options details and explanations.

Prepare the Copy

You first select all source objects, and finally the target object (the target object must be the active object).

It is important that the source objects and the target object are at the same location. If they are placed side-by-side, then the weight transfer will not work. (See the *Vertex Mapping* option.) You can place the objects on different layers, but you have to ensure that all objects are visible when you call the tool.

Now ensure that the target object is in Weight Paint Mode. Open the Toolbar and call the *Transfer Weights* tool in the *Weight Tools* panel.

Adjust Last Operation Panel Confusion

You may notice that the Adjust Last Operation panel stays available after the weight transfer is done. The panel only disappears when you call another Operator that has its own Adjust Last Operation panel. This can lead to confusion when you use Transfer weights repeatedly after you changed your vertex groups. If you then use the still-visible Adjust Last Operation panel, then Blender will reset your work to its state right before you initially called the *Transfer Weights* tool.

So when you want to call the *Transfer Weights* tool again after you made some changes to your vertex groups, then **always** use the *Transfer Weights* button, even if the Adjust Last Operation panel is still available. Unless you really want to reset your changes to the initial call of the tool.

Limit Total

Reduce the number of weight groups per vertex to the specified Limit. The tool removes lowest weights first until the limit is reached.

Hint

The tool can only work reasonably when more than one weight group is selected.

Subset
    Restrict the tool to a subset. See above The Subset Option for how subsets are defined.

Limit
    Maximum number of weights allowed on each vertex.

Set Weight

Reference

Mode:: Weight Paint Mode
Menu:: *Weight ‣ Set Weight*
Shortcut:: `Ctrl-X`

Fill the active vertex group with the current paint weight.

Sample Weight

Reference

Mode:: Weight Paint Mode
Menu:: *Weight ‣ Sample Weight*
Shortcut:: `Shift-X`

Adjust the Weight of the Draw tool to the weight of the vertex under the mouse cursor.

Sample Group

Reference

Mode:: Weight Paint Mode
Menu:: *Weight ‣ Sample Group*
Shortcut:: `Shift-Ctrl-X`

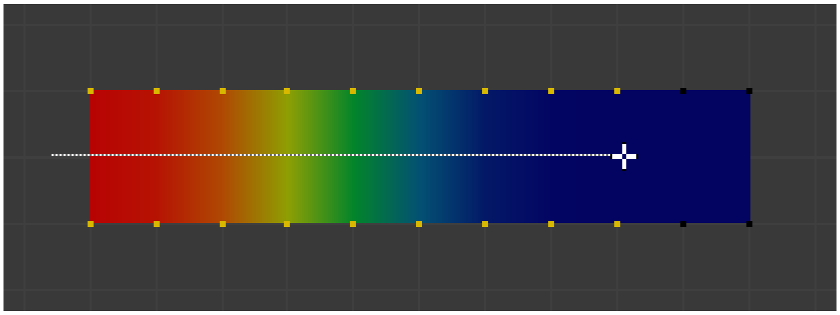
Select one of the vertex groups available under current mouse position.

Gradient (Linear)

Reference

Mode:: Weight Paint Mode
Menu:: *Weights ‣ Gradient (Linear)*
Shortcut:: `Shift-A`

Applies a linear weight gradient; this is useful at times when painting gradual changes in weight becomes difficult. Blends the weights of selected vertices with unselected vertices.



Example of the Gradient tool being used with selected vertices.

Weight
: The gradient starts at the current selected weight value, blending out to nothing.

Strength
: Lower values can be used so the gradient mixes in with the existing weights (just like with the brush).

Type
: The shape of the gradient.

Linear::	Create gradient that forms a straight line.
Radial::	Create gradient that forms a circle.

Gradient (Radial)

Reference

Mode:: Weight Paint Mode

Menu::	*Weights ‣ Gradient (Radial)*
Shortcut::	`Shift-Alt-A`

Applies a radial weight gradient; this is useful at times when painting gradual changes in weight becomes difficult. Blends the weights of selected vertices with unselected vertices.

Weight

The gradient starts at the current selected weight value, blending out to nothing.

Strength

Lower values can be used so the gradient mixes in with the existing weights (just like with the brush).

Type

The shape of the gradient.

Linear::	Create gradient that forms a straight line.
Radial::	Create gradient that forms a circle.

Locks

Reference

Mode::	Edit Mode and Weight Paint Mode
Menu::	*Weights ‣ Locks*
Shortcut::	`K`

Vertex groups can be locked to prevent undesired edits to a particular vertex group.

Tip

Bones that belong to a locked vertex group are displayed in red the 3D Viewport.

Lock All
Locks all vertex groups.

Lock Selected
Locks selected vertex groups.

Lock Unselected
Locks unselected vertex groups.

Lock Only Selected
Lock selected and unlock selected vertex groups.

Lock Only Unselected
Unlock selected and lock unselected vertex groups.

Unlock All
Unlocks all vertex groups.

Unlock Selected
Unlocks selected vertex groups.

Unlock Unselected
Unlocks Unselected vertex groups.

Invert Locks
Inverts the locks on all vertex groups.

Curves Sculpting

- Introduction
    - Curves Menu
    - Selection Modes
    - Select Menu
    - Controls
    - Display
- Brushes
    - Selection Paint
    - Add Curves
    - Delete Curves
    - Density Brush
    - Comb Curves
    - Snake Hook Curves
    - Grow / Shrink Curves
    - Pinch Curves
    - Puff Curves
    - Smooth Curves
    - Slide Curves
- Common Settings
    - Brush
    - Stroke
    - Falloff
    - Cursor

Introduction

Curves Sculpt Mode allows working with curves using various brushes. It is commonly used for hair grooming, but can be used with all kinds of curves.

The curves' surface object plays an important role in many curves sculpting brushes. Most brushes such as Add Curves require the surface to be set already.

Note

Curves Sculpt tools only use the original mesh of the surface object and don't take its modifiers into account.

Curves Menu

Snap to Deformed Surface
: Re-attach curves to a deformed surface using the existing attachment information. This only works when the topology of the surface mesh has not changed.

Snap to Nearest Surface
: Find the closest point on the surface for the root point of every curve and move the root there. This needs to be run after the surface mesh topology changed

Convert to Particle System
: Add a new hair particle system, or update an system on the surface object. The operator is used for backwards compatibility with the old hair type particle system.

Selection Modes

Reference

Mode:: Sculpt Mode
Menu:: *3D Viewport Header ▸ Select Mode*
Shortcut:: `1`, `2`

Selection modes limits selection operators to certain curve domains. This feature is makes it easy to select whole segments at once, or to give more granular control over editing.

Control Points:: Allows selection of individual control points.
Curve:: Limits selection to whole curve segments.

Select Menu

All
: Select all control points or curves.

None
: Deselect all control points or curves.

Invert
: Invert the selection.

Random
: Randomizes inside the existing selection or create new random selection if nothing is selected already.

Endpoints
: Select endpoints of curves. Only supported in the Control Point selection mode.

  Amount Start, End
  : Number of points to select from the front or back of the curve.

Grow
: Select points or curves which are close to already selected elements.

Controls

Sculpt mode provides several properties that give advanced control of the tool’s behavior. These options can be found in the right-hand side of the 3D Viewport’s Header.

Mirror

Allows tools to affect curves symmetrically according to the chosen axis.

Use Sculpt Collision

Prevents the curve segments from passing through the Surface Object.

Display

Overlays

Cage Opacity

Shows the original curves that are currently being edited which is useful with when procedural deformations or child curves are used.

Brushes

Brushes for Curves *Sculpt Mode* bundled in the *Essentials* library.

- Selection Paint
- Add Curves
- Delete Curves
- Density Brush
- Comb Curves
- Snake Hook Curves
- Grow / Shrink Curves
- Pinch Curves
- Puff Curves
- Smooth Curves
- Slide Curves

Selection Paint

Paint curves or control paints to use as masks for the other tools. The selection visibility can be controlled by the *Selection Opacity* option in the Viewport Overlays.

By default the selection sets a new selection. The selection paint can be extended by holding `Shift` and it can be subtracted by holding `Ctrl` while painting.

Brush Settings

Direction

  Determines whether to set a new selection or remove from it. It can be toggled holding `Ctrl` while painting.

Add Curves

Used to distribute new curves on the surface mesh. This tool requires the curve to have a surface object set.

The curves follow the surface normals. Using the interpolation options allows the brush to take the characteristics of existing curves.

Brush Settings

Count
: Number of curves added.

Note

Interpolation allows to add hair which are already combed. The new curves are created following the previously created curves which are in the vicinity.

Interpolate Length
: Use the average length of the curves in close proximity.

Interpolate Radius
: Use the average radius of the curves in close proximity. If there is no radius attribute, then the interpolation will skip.

Interpolate Shape
: Use the average shape of the curves in close proximity.

Interpolate Point Count
: Use the average amount of control points of the curves in close proximity.

Curve Length

Length of newly added curves when not interpolated.

Curve Radius

Radius of newly added curves when not interpolated.

Points per Curve

Number of Control Points for the new created curves when the point count is not interpolated.

Delete Curves

Remove existing curves. The tool deletes the entire curves, if any of its segments fall under the brush falloff radius.

Density Brush

Create (or remove) curves based on a target distance. It generates a high number of points and then rejects the ones that are too close to existing points.

Brush Settings

Density Mode

Determines whether the brush adds or removes curves.

Auto::	Either add or remove curves depending on the distance between existing curve roots under the cursor.
Add::	Add new curves between existing curves, taking the minimum distance into account.
Remove::	Remove curves whose root points are too close.

Distance Min

Goal distance between the curve roots.

Edit Minimum Distance R

Interactively sets the *Distance Min* value by displaying a graphic inside the brush cursor, giving a representation of the density.

The density can be adjusted by moving the mouse cursor closer or farther from the paint cursor. The *Distance Min* will be changed once the operator is confirmed.

Count Max

The maximum amount of points that the brush tries to sample in the surface.

Comb Curves

Shape the curves by moving their control points while preserving the initial length of every curve segment.

Brush Settings

Curve Falloff
   Falloff that is applied from the tip to the root of each curve.

Snake Hook Curves

Extend existing curves in a specific direction of the brush strokes. A curve can only be extended by its tips (the control point opposite to the root).

Note

No new control points are created for the curves. Instead, existing points are sampled along the new curve shape.

Grow / Shrink Curves

Change the length of existing curves preserving the amount of control points and resampling the curve to preserve the original shape.

Brush Settings

Direction
: Determines whether to grow or shrink the curves. It can be toggled holding `Ctrl` while sculpting.

Scale Uniform
: Grow or shrink curves by changing their size uniformly instead of using trimming or extrapolation.

Minimum Length
: Avoid shrinking curves shorter than this length.

Pinch Curves

Converges adjacent curves to the location at the center of the cursor.

The pinch brush can be inverted by holding `Ctrl`.

Puff Curves

Makes the curves stand up. The brush aligns the curve with the surface normal and makes sure that points don’t move closer to the root point.

Smooth Curves

This brush makes curve segments close to one another more parallel.

Slide Curves

Slides the curves along the surface mesh. This tool requires the curve to have a Surface. Each curve is also rotated by the change in the surface normal.

Common Settings

Information on brush settings for every mode can be found in these pages:

Brush

See general and advanced Brush here.

Stroke

See the global brush settings for Stroke settings.

Falloff

See the global brush settings for Falloff settings.

Cursor

See the global brush settings for Cursor settings.

Grease Pencil

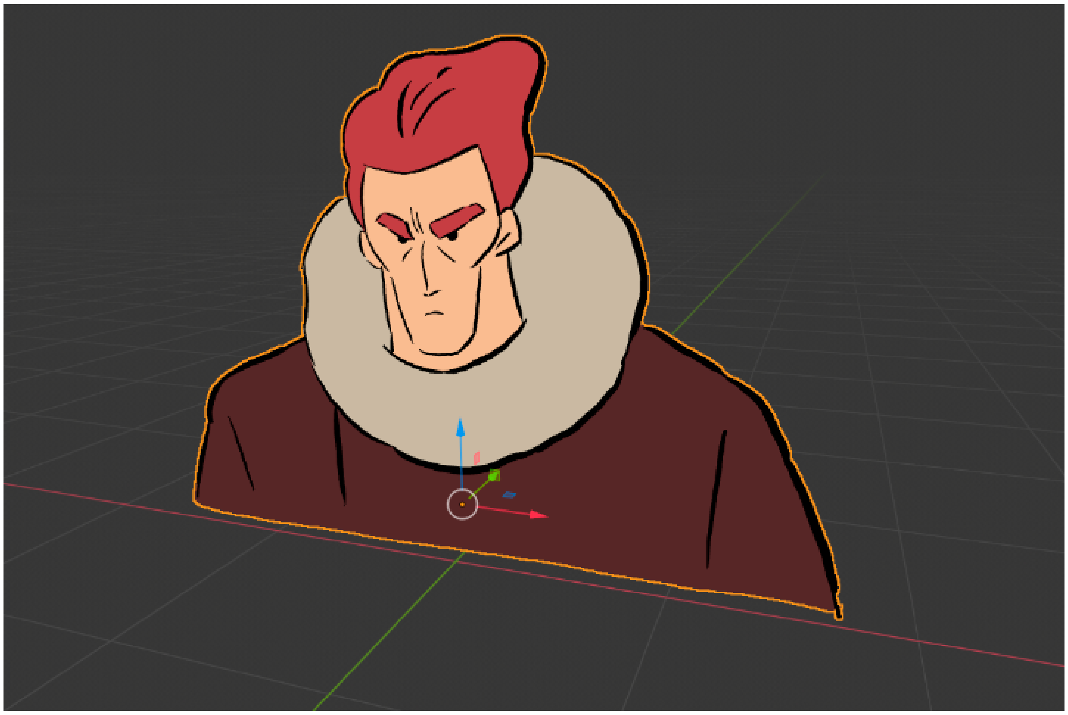
- Introduction
    - Quick Start
- Structure
    - Points
    - Edit Lines
    - Strokes
- Primitives
    - Blank
    - Stroke
    - Monkey
    - Scene Line Art
    - Collection Line Art
    - Object Line Art
- Properties
    - Object Properties
    - Data Properties
- Modifiers
    - Introduction
    - Generate
    - Deform
    - Color
    - Edit
- Visual Effects
    - Introduction
    - Types
- Materials
    - Material Shader
    - Setting Up Materials
    - Properties
- Multiframe
    - Usage
- Animation
    - Introduction

  - Interpolation
  - Animation Tools
- Object Modes
  - Draw Mode
  - Sculpt Mode
  - Edit Mode
  - Vertex Paint Mode
  - Weight Paint Mode
  - Object Mode

Introduction

Grease Pencil is a Blender object. It accepts the drawing information from a mouse or pressure-sensitive stylus and places it in 3D space as a collection of points, which are defined as a stroke.

The Grease Pencil object can be used to make traditional 2D animation, cut-out animation, motion graphics, or used it as storyboard tool, among other things.



An illustration in 3D space using the Grease Pencil object.

Strokes are created in Draw Mode, which requires a new keyframe in the animation timeline for the Grease Pencil object. Existing strokes can then be adjusted in Edit Mode and Sculpt Mode. Finally, artists can apply materials, modifiers, lighting, and visual effects to strokes.

Quick Start

Artists can add Grease Pencil to any existing Blender scene, or start with a 2D Animation template. The template offers some pre-configured options that are helpful for animation and storyboarding.

Create and Use Grease Pencil

1. From Object Mode, *Add* ‣ *Grease Pencil* ‣ *Blank*.
2. Create a new keyframe or turn on Auto Key. (See Keyframe Editing)
3. Switch to Draw Mode.
4. Click and drag across the viewport to add strokes to the Grease Pencil object.

2D Animation Template

To create a new Blender file using the “2D Animation” project template use: *File* ‣ *New* ‣ *2D Animation*.

Note the following pre-configured setup for the 2D Animation template:

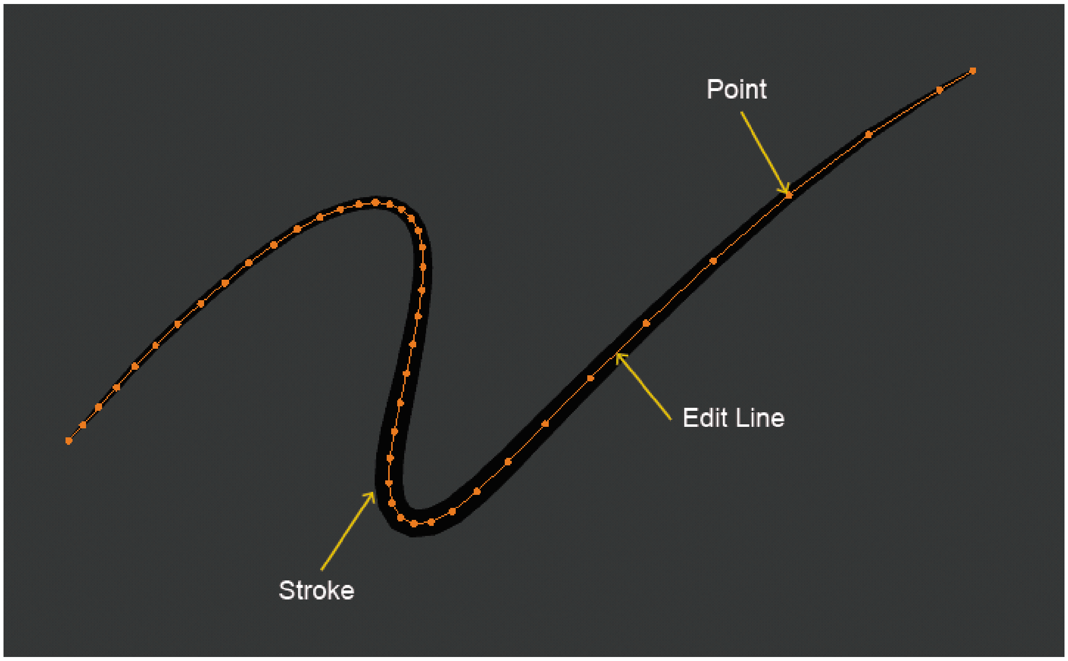
- 2D Animation is the default active workspace.
- *World Properties* ‣ *Surface (Background)* ‣ *Color* is set to white.
- *Render Properties* ‣ *Color Management* is set to Standard.
- The drawing plane is set to Front (X-Z).
- Line and Fill layers, along with some stroke materials, are configured for Grease Pencil.
- The animation timeline will automatically create a new keyframe when Grease Pencil is used on empty frames.

Tip

Grease Pencil can read pressure-sensitivity information from a Graphics Tablet or stylus.

Structure

Grease Pencil object has three main basic components: *points*, *edit lines* and *strokes*.



Example of Grease Pencil structure.

Points

The main element used in editing Grease Pencil objects are points. Points represent a single point in 3D space.

Each point stores all the properties that define the final appearance of the strokes as its location, thickness, alpha, weight and UV rotation for textures.

Note

Point (Grease Pencil) and Vertex (meshes) are equivalent names.

Edit Lines

Points are always connected by a straight line, which you see when you are editing in Edit Mode or when you look at a stroke in wireframe view. They are invisible on the rendered image and are used to construct the final stroke.

Strokes

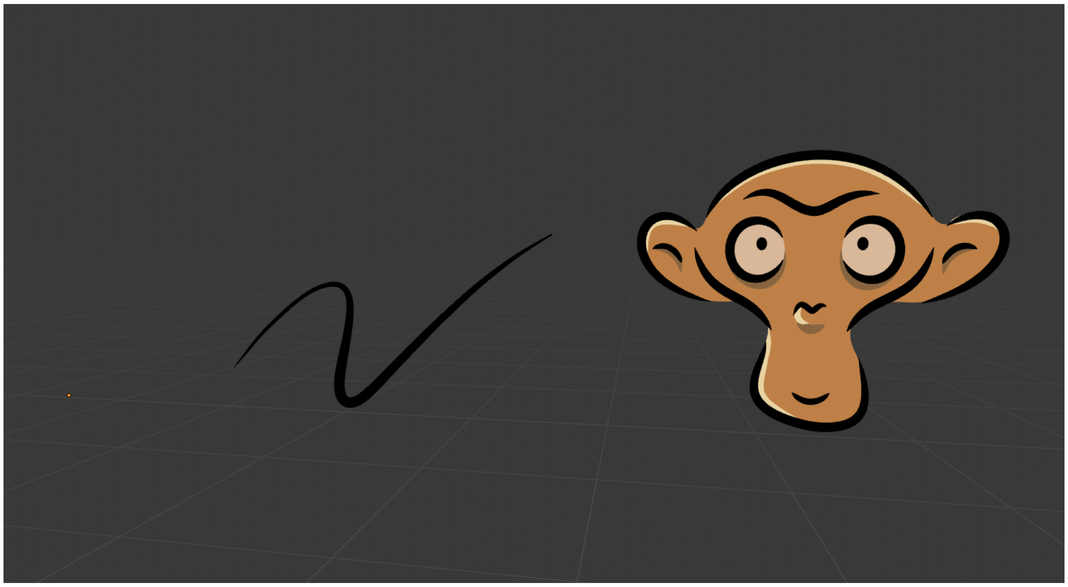
The stroke is the rendered image of the points and edit lines, using a particular Grease Pencil material. (Grease Pencil materials are linked at stroke level.)

Primitives

Reference

Mode::	Object Mode and Edit Mode
Menu::	*Add* ‣ *Grease Pencil*
Shortcut::	`Shift-A`

In Object Mode, the *Add* menu provides three different Grease Pencil primitives with preset materials and 2D layers:



Grease Pencil primitives.

Blank

Adds a Grease Pencil object without any stroke.

Stroke

Adds a Grease Pencil object with a simple stroke as a reference.

Monkey

It creates a 2D monkey head. The Monkey’s name is “Suzanne” and is Blender’s mascot. 2D Suzanne is very useful as a standard test.

Scene Line Art

Sets up a Line Art Modifier for the active scene by creating an “empty” Grease Pencil object with a Line Art modifier referencing each object in the scene.

Collection Line Art

Sets up a Line Art Modifier for the active collection by creating an “empty” Grease Pencil object with a Line Art modifier referencing each object in the collection.

Object Line Art

Sets up a Line Art Modifier for the active object by creating an “empty” Grease Pencil object with a Line Art modifier referencing the active object.

Properties

- Object Properties
  - Visibility
- Data Properties
  - Layers
  - Onion Skinning
  - Settings
  - Attributes
  - Vertex Groups
  - Custom Properties

Object Properties

Visibility

Reference

Panel::	*Object Properties ‣ Visibility*

Use Light

Enables the Grease Pencil object to be affected by lights.

This property affect the whole object, for more control with lights you can enable or disable the use of lights by layers. See Layers for more information.

Lights disabled (left) and enabled (right).

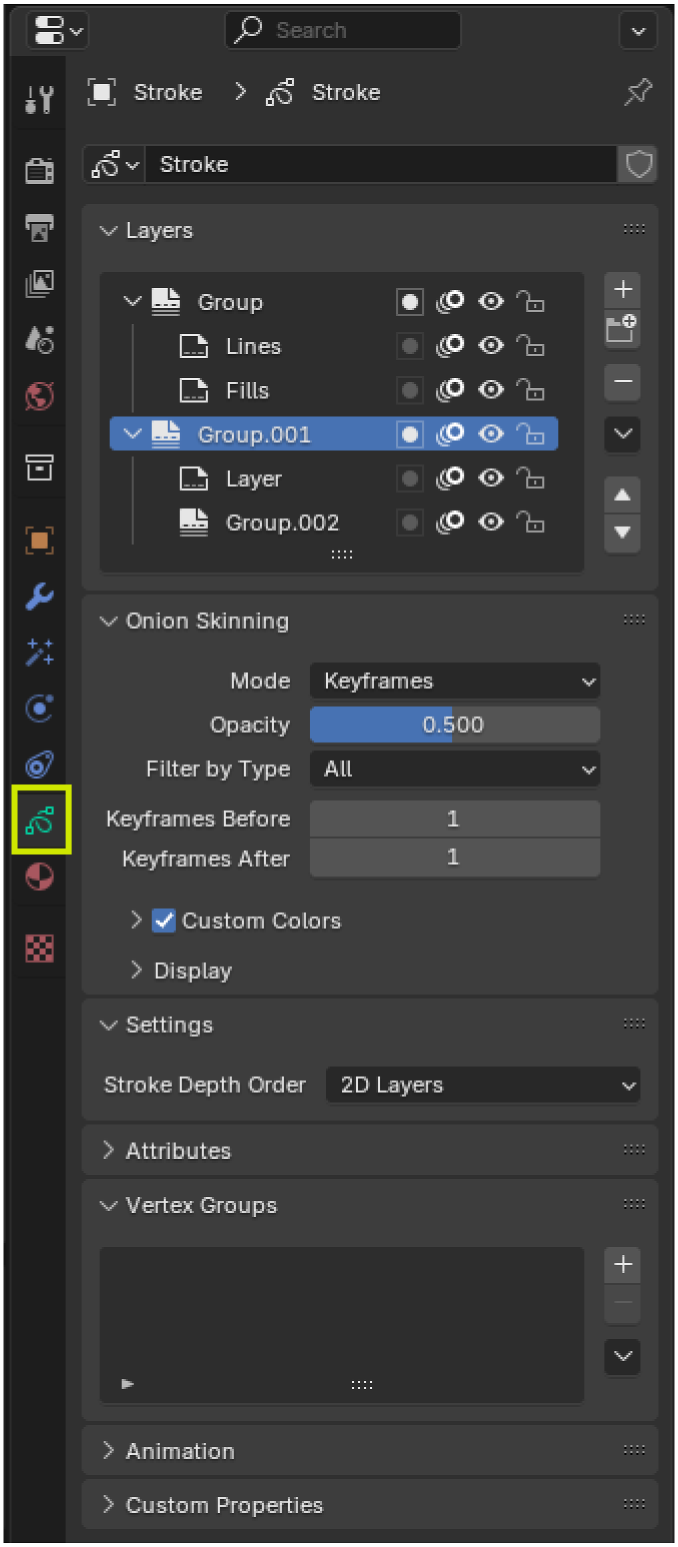
See also

There are several other general visibility properties.

Data Properties

Grease Pencil

The Grease Pencil data-block menu can be used to link the data between objects.


Search
Stroke
Stroke
Stroke
Layers
Group
Lines
Fills
Group.001
Layer
Group.002
Onion Skinning
Settings
Attributes
Vertex Groups
Animation
Custom Properties

Mode Keyframes

Opacity 0.500

Filter by Type All

Keyframes Before 1

Keyframes After 1

☑ Custom Colors

Display

Stroke Depth Order 2D Layers

Grease Pencil Object Data.

Layers

Strokes can be grouped in 2D layers, a special Grease Pencil layers that help to organize the drawing order and visibility of the strokes. Layers can be organized into layer groups.

Onion Skinning

Onion skinning is used in animation to see several frames at once and make decisions or edits based on how the previous/next frames are drawn.

Settings

General settings for Grease Pencil strokes.

Attributes

Layers can store Custom Attributes. The attributes are stored on the Layer domain.

For example, the Layer Adjustments are stored as layer attributes.

Attributes

List view of all the attributes stored on the layers.

Name

Name of the layer attribute.

Data Type

The Data Type of the attribute.

Vertex Groups

Vertex groups can be used to assign a group or weighted group to some operator. An object can have several weight groups and can be assigned in

Weight Paint Mode.

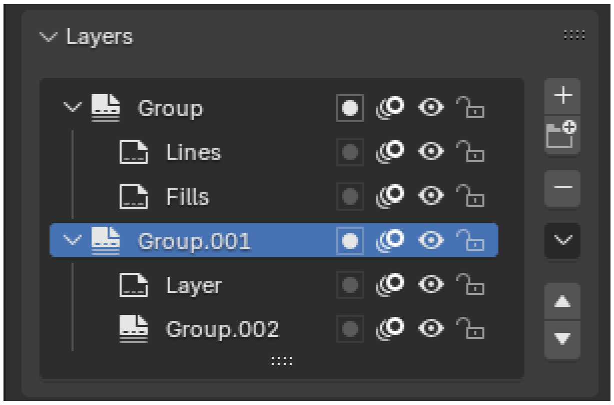
Custom Properties

Create and manage your own properties to store data in the Grease Pencil's data-block.

Layers

Reference

Mode:: All Modes
Panel:: *Object Data tab ▸ Layers*



Grease Pencil Layers panel.

Grease Pencil objects can be organized into a tree known as the layer tree for grouping and arranging strokes.

Any stroke can only belong to a single 2D layer. The selected layer is the active layer. Only one layer or group can be active at a time. When you draw, the new strokes are added to the active layer. By default the view order of the layers in the viewport is top to bottom.

Layers can be grouped using Layer Groups. A layer can only be in one group at a time. Layers can be moved into groups using drag-and-drop. Groups can be color coded with a color tag.

Every layer correspond to a channel in the Dope Sheet editor (in Grease Pencil mode). See Dope Sheet for more information.

Layers can also be used together with Modifiers to only affects part of your drawing. See Modifiers for more information.

Layers can mask other layers by enabling Use Mask (mask icon) or using the checkbox in the Masks panel header. See Masks for more information.

Tip

Sometimes the layers you are not working on can be a distraction in the 3D Viewport. Activate the Fade Inactive Layers overlay to control the opacity of the non-active layers.

Layer Tree

Tree view of all layers and groups for the Grease Pencil object.

Next to the layer name there are four icons buttons that control common properties of the layer:

Use Mask (mask icon)

Toggle the affect of Masks on the layer.

Onion Skinning (onion skin icon)

Toggle using the layer for Onion Skinning.

Hide (eye icon)

Toggle layer visibility in the viewport and in render.

Lock (padlock icon)

Toggle layer from being editable.

Add New Layer

Adds a new layer to the active object.

Add New Layer Group

Adds a new layer group to the active object. Note, layer groups cannot be added from the Dopesheet; they must be added from the Properties editor.

Remove Layer/Group

Removes the active layer or layer group.

Layer Specials

Operators for working with layers.

Duplicate

Makes an exact copy of the selected layer appending a number to differentiate its name.

Duplicate Empty Keyframes

Makes a copy of the selected layer but with empty keyframes. Useful to easily have empty keyframes preset to work on the cleanup or filling process.

Show All

Turns on the visibility of every layer in the list.

Hide Others

Turns off the visibility of every layer in the list except the active one.

Lock All

Locks edition of all the layers in the list.

Unlock All

Unlocks edition of all the layers in the list.

Autolock Inactive Layer

Locks automatically the edition of every layer in the list except the active one. This way you avoid to make unwanted changes in other layers without the need to lock them every time.

Use Locked Materials Editing

Avoids editing locked materials in the layer. When disabled, any material can be edited even if they are locked in the material list.

Merge Down `Shift-Ctrl-M`

Combine the selected layer with the layer below, the new layer keeps the name of the lower layer.

Merge Group

Combine layers in the active layer group into a single layer.

Merge All

Combine all layers into the active layer.

Copy Layer to Selected
Copy the active layer to the selected Grease Pencil object.

Copy All Layers to Selected
Copy all layers to the selected Grease Pencil object.

Reorder Layer
Moves the active layer or layer group up/down in the tree.

Below the layers list there are additional settings:

Blend Mode
The layer blending operation to perform. See Color Blend Modes.

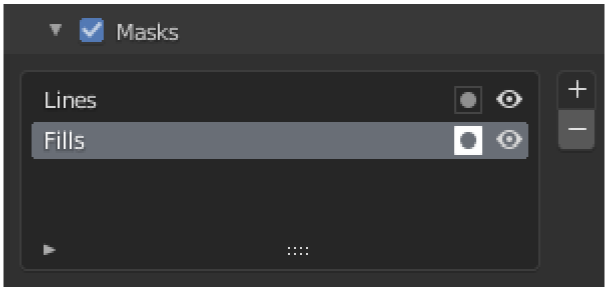
Opacity
Used to set the opacity of the layer.

Lights
When enabled, the layer is affected by lights.

Masks

In Grease Pencil there are no special mask layers, any layer can act as a mask for other layers. The mask system is flexible enough to allow top-bottom and bottom-top masking.

Layers used as masks can use all the blend modes and different opacity values like any other layer.



Masks panel.

Tip

If you want to make a full transparent masking you will have to set the mask layer’s opacity to 0.

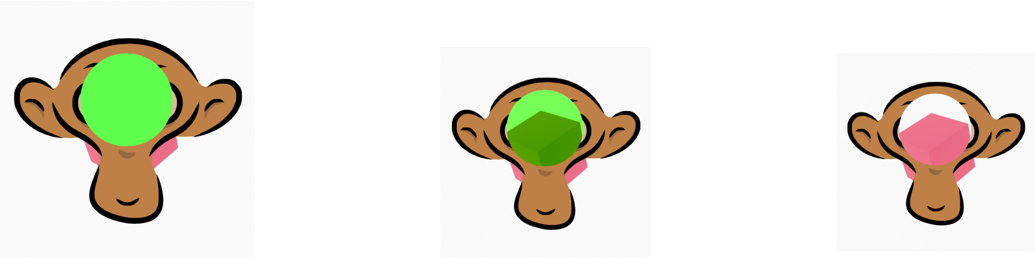
The layer/s that will act as mask of the current layer could be added to the Mask list view.

In the Masks list next to the layers name there are two icons buttons that control common properties of the layer mask:

Invert (mask icon)
Inverts the mask.

Viewport/Render Visibility (eye icon)
Toggle layer visibility in the viewport and in render.

Mask (green circle) samples.



Original image (Blend: Regular, Opacity: 1).

Blend: Hard Light, Opacity: 1.

Blend: Regular, Opacity: 1.

Transform

Allows per-layer location, rotation and scale transformations.

Adjustments

Tint Color

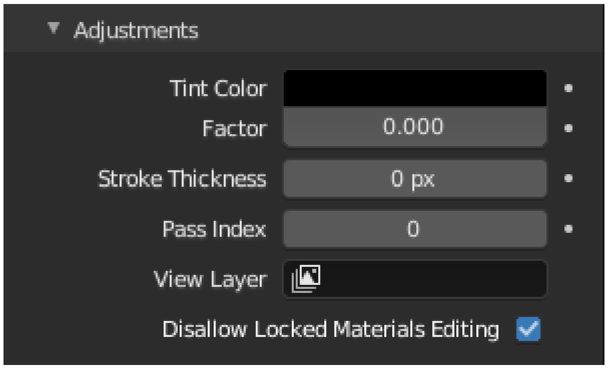
Color that tint any material colors used in the layer.

Factor

Controls the amount of tint color to apply.

Stroke Thickness

Thickness value that override the strokes thickness in the layer.



Adjustments

Tint Color

Factor 0.000

Stroke Thickness 0 px

Pass Index 0

View Layer

Disallow Locked Materials Editing ☑

Layers adjustment panel.

Relations

Parent

Select a Parent object to manipulate the layer. The layer will inherit the transformations of the parent, this is especially useful when rigging for cut-out animation.

Pass Index

The layer index number can be used with some modifiers to restrict changes to only certain areas.

See Modifiers for more information.

View Layer

Defines the View Layer to use for the Grease Pencil layer. If empty, the layer will be included in all View Layers. This is useful to separate drawings parts for compositing.

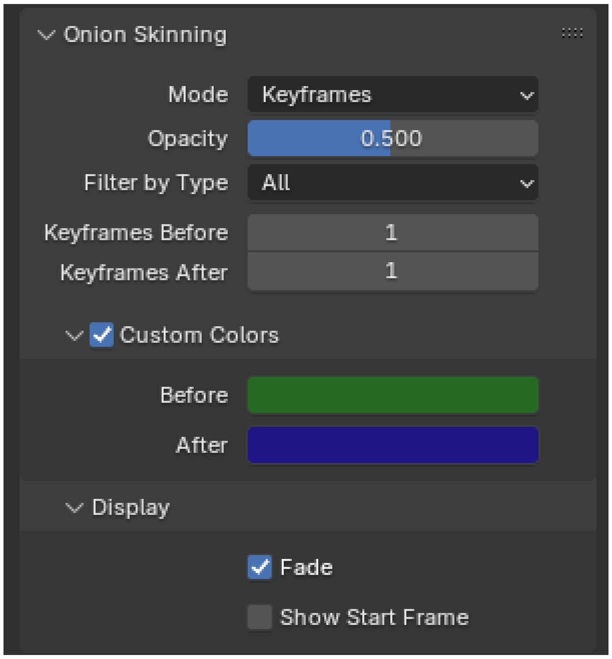
Use Masks in Render

If disabled, no masks on the layer are included in the view layer render.

Onion Skinning

Onion Skinning show ghosts of the keyframes before and after the current frame allowing animators to make decisions in the animation sequence.

The main switch to show/hide Onion Skinning is in the Viewport Overlays, but Grease Pencil Onion Skinning is per-layer and the visibility can be toggle in the layer list. See 2D Layers for more information.



Onion Skinning

Mode: Keyframes
Opacity: 0.500
Filter by Type: All
Keyframes Before: 1
Keyframes After: 1

☑ Custom Colors

Before: 🟩
After: 🟦

Display

☑ Fade
☐ Show Start Frame

Onion Skinning panel.

Options

Mode

Keyframes::	Shows Keyframes in the range determined by the *Before/After* settings.
Frames::	Shows Frames in the range determined by the *Before/After* settings.
Selected::	

Shows only on the manually selected keyframes in the Dope Sheet.

Opacity
Control the opacity of the ghost frames.

Filter by Type
Filters what type of frames to show in the Onion Skinning range.

Keyframes Before/After
Sets how many frames or keyframes, depending on the *Mode*, to show before and after the current frame.

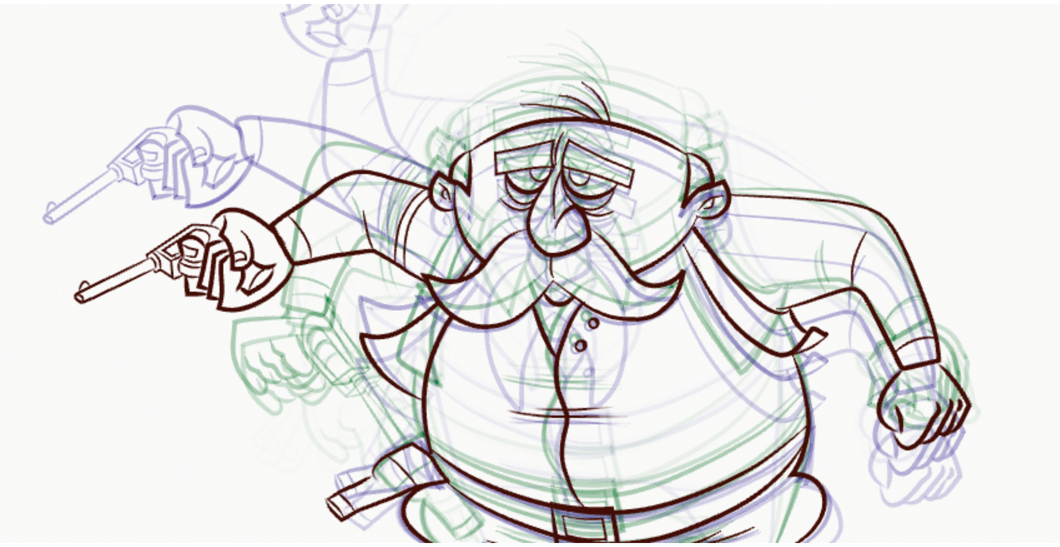
Custom Colors

Before/After
Color to use before and after the current frame on ghost frames.

Display

Fade
Opacity of the ghosts frames decrease the further away from the current frame.

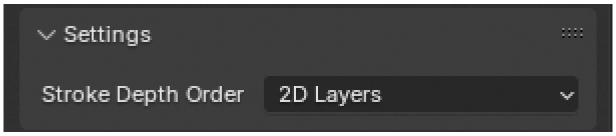
Show Start Frame
Help working on loop animations showing the first keyframe/frame as ghost when you are on the last frame of your animation.



An example of Onion Skinning activated.

Settings

General settings for Grease Pencil strokes.



Settings

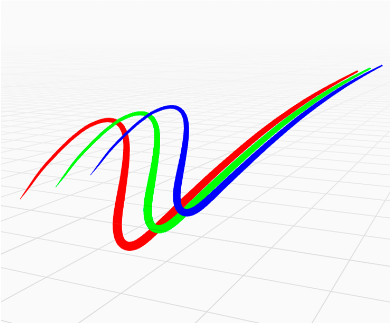
Stroke Depth Order 2D Layers

Strokes panel.

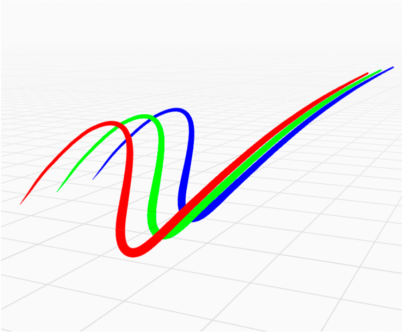
Stroke Depth Order

Defines how the strokes are ordered in 3D space (for objects not displayed *In Front*).

2D Layers::	The Strokes drawing order respect the order of the 2D layers list (top to bottom) and ignores the real position of the strokes in 3D space. See 2D Layers for more information.
3D Location::	The strokes drawing order is based on the stroke location in 3D space.



Blue, Green and Red strokes in three different layers using 2D Layers depth order.



Blue, Green and Red strokes in three different layers using 3D Location depth order.

Grease Pencil Modifiers

- Introduction
  - Interface
- Generate
  - Array Modifier
  - Build Modifier
  - Dot Dash Modifier
  - Envelope Modifier
  - Length Modifier
  - Line Art Modifier
  - Mirror Modifier
  - Multiple Strokes
  - Outline Modifier
  - Simplify Modifier
  - Subdivide Modifier
- Deform
  - Armature Modifier
  - Hook Modifier
  - Lattice Modifier
  - Noise Modifier
  - Offset Modifier
  - Shrinkwrap Modifier
  - Smooth Modifier
  - Thickness Modifier
- Color
  - Hue/Saturation Modifier
  - Opacity Modifier
  - Tint Modifier
- Edit
  - Texture Mapping Modifier
  - Time Offset Modifier
  - Vertex Weight Angle Modifier
  - Vertex Weight Proximity Modifier

Introduction

Reference

Panel:: *Properties* ▸ *Modifiers*

Grease Pencil has their own set of modifiers. Modifiers are automatic operations that affect an object in a non-destructive way. With modifiers, you can perform many effects automatically that would otherwise be too tedious to do manually and without affecting the base geometry of your object.

With Geometry Nodes, it is possible to create custom Grease Pencil modifiers.

They work by changing how an object is displayed and rendered, but not the geometry which you can edit directly. You can add several modifiers to a single object forming the modifier stack and Apply a modifier if you wish to make its changes permanent.

There are four types of modifiers for Grease Pencil:

Edit
: These are tools similar to the Deform ones (see below), however, they usually do not directly affect the geometry of the object, but some other data, such as vertex groups.

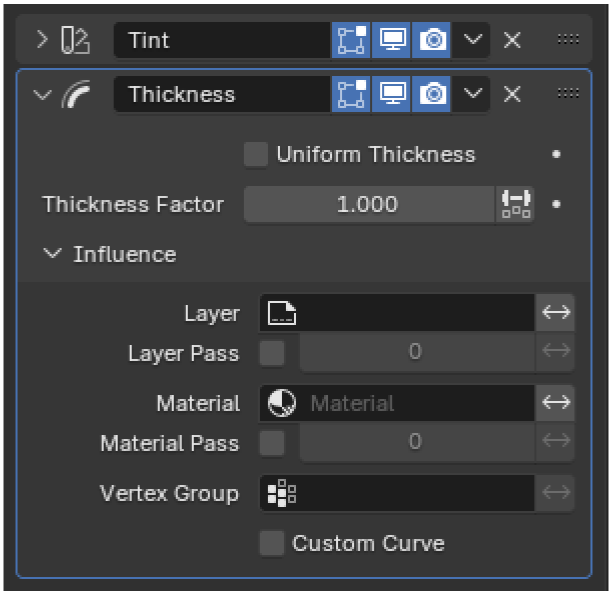
Generate
: The *Generate* group of modifiers includes constructive tools that either change the general appearance of or automatically add new geometry to an object.

Deform
: The *Deform* group of modifiers only changes the shape of an object without adding new geometry,

Color

The *Color* group of modifiers change the object color output.

Interface


Tint
Thickness
Uniform Thickness
Influence
Layer
Material
Vertex Group
Custom Curve

Thickness Factor 1.000

Layer Pass 0

Material Pass 0

Panel layout (Thickness modifier as an example).

Each modifier's interface shares the same basic components like modifiers for meshes.

See [Modifiers Interface]() for more information.

Note

Grease Pencil strokes, unlike meshes, still can not be edited directly in the place.

Applying Modifiers

Applying a modifier makes the effects of the modifier "real"; converts the strokes to match the applied modifier's results, and deletes the modifier.

When applying a modifier to an object that shares Object Data between multiple objects, the object must first be made a Single User which can be performed by confirming the pop-up message.

Warning

Applying a modifier that is not first in the stack will ignore the stack order (it will be applied as if it was the first one), and may produce undesired results.

Reference

Panel::	*Properties* ▸ *Modifiers* ▸ *Modifier Header* ▸ *Specials*

Apply (Active Keyframe) `Ctrl-A`
Applies the modifier for the current keyframe.

Apply (All Keyframes)
Applies the modifier for all keyframes.

Note

With Geometry Nodes it is possible to add new layers to the geometry. When applying, this will create a single keyframe on the first frame of evaluation. Layers with duplicated names in evaluated geometry will be deduplicated.

It is also possible to have layers with empty names. When applying these get renamed to *Layer* (and *Layer.001* etc. when such a layer already exists in the original geometry).

Influence Filters

Most of the modifiers share some special properties that restrict the effect only to certain items.

Layer
: Restricts the effect only to one layer or to any layers that share the same material Pass Index.

Material
: Restricts the effect only to material that share the same material or material Pass Index.

Vertex Group
: Restricts the effect only to a vertex group.

Custom Curve
: When enabled, use a custom curve to shape the effect along the strokes from start to end points.

The Invert toggle <–> allows you to reverse the filters behavior.

Grease Pencil Generate Modifiers

- Array Modifier
- Build Modifier
- Dot Dash Modifier
- Envelope Modifier
- Length Modifier
- Line Art Modifier
- Mirror Modifier
- Multiple Strokes
- Outline Modifier
- Simplify Modifier
- Subdivide Modifier

Array Modifier

The *Array* modifier creates an array of copies of the base object, with each copy being offset from the previous one in any of a number of possible ways.

Useful for creating complex repetitive drawings.

Multiple Array modifiers may be active for an object at the same time (e.g. to create complex three-dimensional constructs).

See also

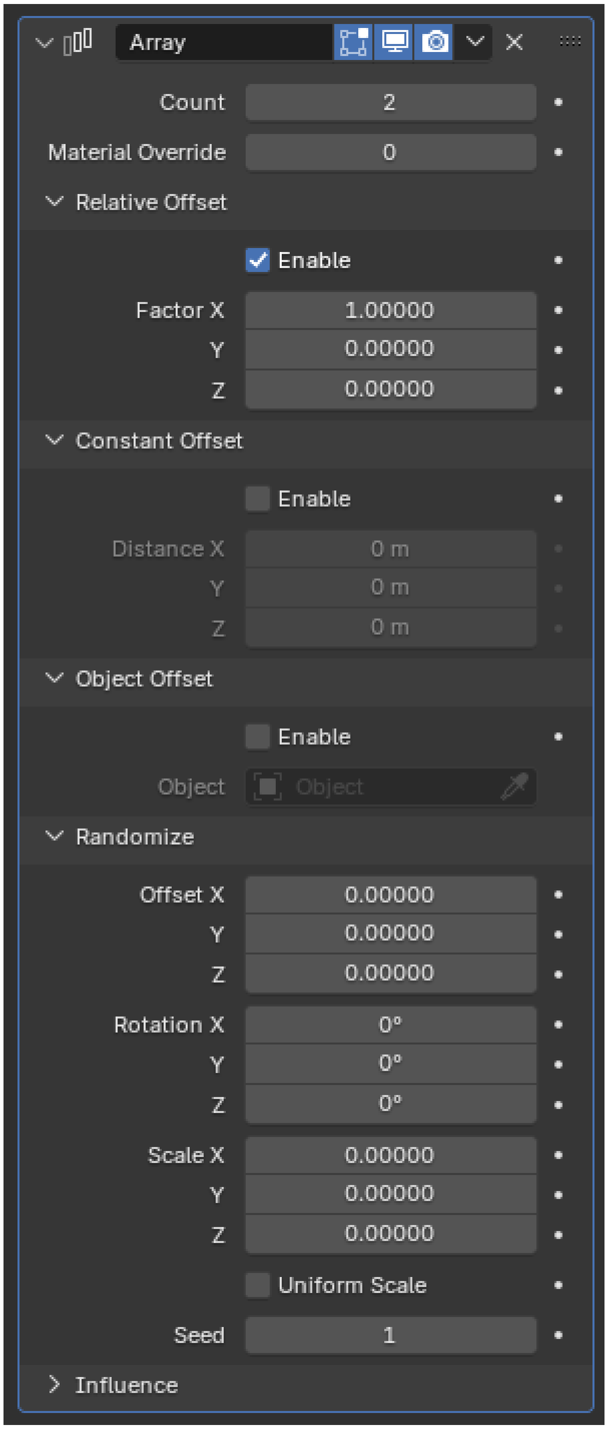
This documentation refers to the Array Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Array Modifier.

Options

Count
: Total number of copies.

Material Override
: Index of the material to use on duplicated strokes (0 use strokes original materials).

Relative Offset


Array

Count: 2

Material Override: 0

Relative Offset

☑ Enable

Factor X: 1.00000

Y: 0.00000

Z: 0.00000

Constant Offset

☐ Enable

Distance X: 0 m

Y: 0 m

Z: 0 m

Object Offset

☐ Enable

Object: Object

Randomize

Offset X: 0.00000

Y: 0.00000

Z: 0.00000

Rotation X: 0°

Y: 0°

Z: 0°

Scale X: 0.00000

Y: 0.00000

Z: 0.00000

☐ Uniform Scale

Seed: 1

Influence

The Array modifier.

Relative Offset

Factor X, Y, Z

Adds a translation equal to the object's bounding box size along each axis, multiplied by a scaling factor, to the offset. X, Y and Z scaling factors can be specified.

Constant Offset

Factor X, Y, Z

Adds a constant translation component to the duplicate object's offset. X, Y and Z constant components can be specified.

Object Offset

Distance X, Y, Z

Adds a transformation taken from an object (relative to the current object) to the offset. It is good practice to use an empty object centered or near to the initial object.

Randomize

Offset X, Y, Z

Add random offset values to the copies.

Rotation X, Y, Z

Add random rotation values to the copies.

Scale X, Y, Z

Add random scale values to the copies.

Uniform Scale

Use the same random *Seed* for each scale axis in the copies for a uniform scale.

Seed

Seed used by the pseudo-random number generator.

Note

The *Depth Order* used in the Grease Pencil object has an influence on the visualization of the strokes when using the Array modifier. See Depth Order for more information.

Influence

See Influence Filters.

Build Modifier

The *Build* modifier make strokes appear or disappear in a frame range to create the effect of animating lines being drawn or erased.

See also

This documentation refers to the Build Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Build Modifier.

Options

Mode

Determines how many strokes are being animated at a time.

Strokes appear/disappear one after the other, but only a single one changes at a time.

Multiple stroke appear/disappear at a time.

Builds only the strokes that are new compared to last keyframe. The assumption is Additive Drawing was used so that the shared strokes are the same.

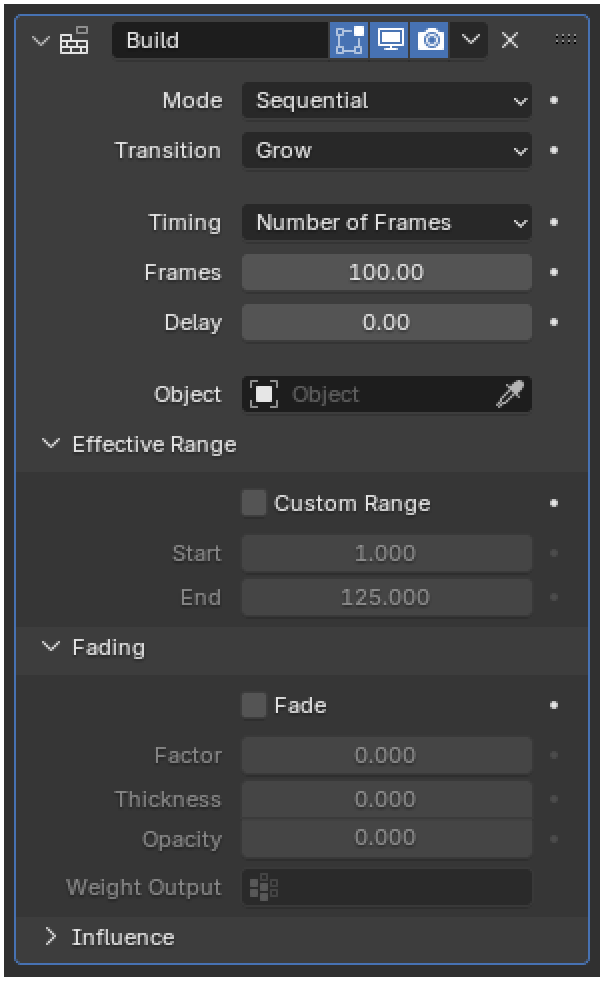
Transition *(in Sequential and Concurrent Mode)*

Determines the animation type to build the strokes.

Shows points in the order they occur in each stroke, from first to last stroke. (Simulating lines being drawn.)

Hide points from the end of each stroke to the start, from last to first stroke. (Simulating lines being erased.)

Sequential::
Concurrent::
Additive::
Grow::
Shrink::
Vanish::


Build
Mode: Sequential
Transition: Grow
Timing: Number of Frames
Frames: 100.00
Delay: 0.00
Object: Object

☐ Custom Range
Start: 1.000
End: 125.000

☐ Fade
Factor: 0.000
Thickness: 0.000
Opacity: 0.000

The Build modifier.

Hide points in the order they occur in each stroke, from first to last stroke. (Simulating ink fading or vanishing after getting drawn.)

Timing

The way you want to time the building of the strokes.

Natural Drawing Speed::

Use the recorded speed of the stylus when the strokes were drawn. Only available in Sequential and Additive Mode.

Speed Factor

The recorded speed is multiplied by this value.

Maximum Gap

The maximum gap between strokes in seconds.

Number of Frames::

Set a fixed maximum number of frames for the build animation. (Unless another Grease Pencil keyframe occurs before this time has elapsed.)

Frames

The maximum number of frames used.

Delay

Number of frames after each Grease Pencil keyframe before the modifier has any effects.

Percentage Factor:: Manually set a percentage factor to control the amount of the strokes that are visible.

Factor

The factor from 0 to 1.

Time Alignment:: Only available in Concurrent Mode.

Align Start

All stroke start at the same time (i.e. shorter strokes finish earlier).

Align End

All stroke end at the same time (i.e. shorter strokes start later).

Object

Use the distance to an object to define the order in which strokes appear.

Custom Range

If enabled, only modify strokes during the specified frame range.

Start, End

Determines the start and end frame for the build effect.

Fade

Factor
Defines how much the stroke is fading in/out.

Thickness
How much strength fading is applied to the stroke's thickness.

Opacity
How much strength fading applies to the stroke's opacity.

Weight Output
Assign a weight value to points that have started/finished the fade.

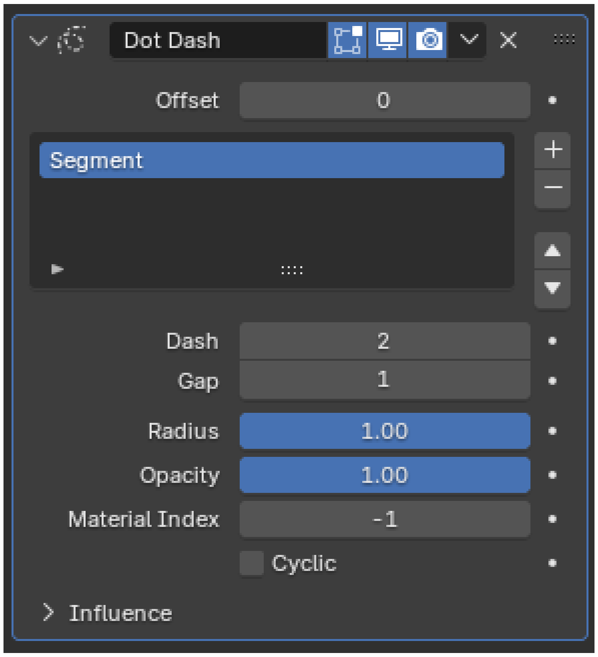
Influence Filters

See Influence Filters.

Dot Dash Modifier

The *Dot Dash* modifier generates dot dash segments from the original stroke.

Options



Offset 0

Segment

Dash 2

Gap 1

Radius 1.00

Opacity 1.00

Material Index -1

☐ Cyclic

The Dot Dash modifier.

Offset

Determines the starting offset of the pattern.

Segment

Makes up individual stroke of a dot dash pattern.

Use the plus/minus button on the side of the list to add/remove segments.

Dash

The number of consecutive points from the original stroke to include in this segment.

Gap

The number of points skipped after the segment ends.

Radius

The factor to apply to the original point's radius for the new points.

Opacity

The factor to apply to the original point's opacity for the new points.

Material Index
    Use this index on generated segment, use -1 for existing material.

Use Cyclic
    Close the segment.

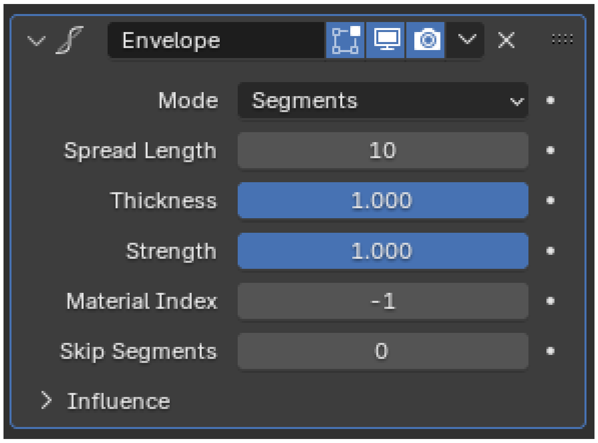
Influence Filters

See Influence Filters.

Envelope Modifier

The *Envelope* modifier creates a shape known as envelope over the existing strokes connecting all the points that have n points between them.

Options



Mode: Segments
Spread Length: 10
Thickness: 1.000
Strength: 1.000
Material Index: -1
Skip Segments: 0

The Envelope modifier.

Mode

Deform::	Replaces the original stroke with the envelope shape.
Segments::	Add segments to create the envelope shape keeping the original stroke.
Fill::	Add segments to create the envelope without the original stroke.

Spread Length
    The number of points to skip when creating the straight segments that define the envelope.

Thickness
    The thickness of the generated stroke segments.

Strength
: The Opacity of the generated stroke segments.

Material Index
: Defines the material to use on the generated stroke segments.

Skip Segments
: The Number of generated stroke segments to skip to reduce complexity.

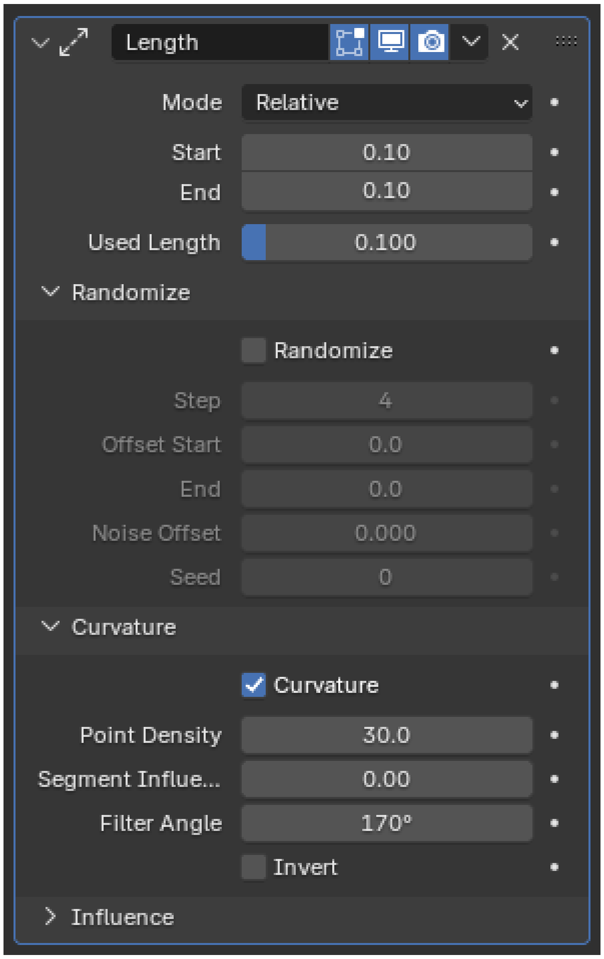
Influence Filters

See Influence Filters.

Length Modifier

The *Length* modifier can shrink or extend strokes.

Options



Length

Mode: Relative

Start: 0.10

End: 0.10

Used Length: 0.100

Randomize

☐ Randomize

Step: 4

Offset Start: 0.0

End: 0.0

Noise Offset: 0.000

Seed: 0

Curvature

☑ Curvature

Point Density: 30.0

Segment Influe...: 0.00

Filter Angle: 170°

☐ Invert

Influence

The Length modifier.

Mode

Absolute:: Length is in geometry space.

Relative:: Length is in ratio to the stroke's length.

Start

Added length to the start of the stroke. Negative value will shrink the stroke.

End

Added length to the end of the stroke. Negative value will shrink the stroke.

Used Length

Define what portion of the stroke is used to calculate the direction of the extension.

Curvature

When enabled, the extension will follow the curvature of the stroke.

Point Density

Multiplied by Start/End for the total point count.

Segment Influence

Factor to determine how much the length of the individual segments should influence the final computed curvature. Higher factors makes small segments influence the overall curvature less.

Filter Angle

Ignore points on the stroke that deviate from their neighbors by more than this angle when determining the extrapolation shape.

Invert

Invert the curvature of the stroke's extension.

Random Offsets

Random Offset Start/End

Size of random length added to the start/end of each stroke.

Random Noise Offset

Smoothly offset each stroke's random value.

Seed
: Number used to generate different noise patterns.

Randomize

Re-randomizes values over time.

Step
: Number of frames before recalculate random values again.

Influence Filters

See Influence Filters.

Line Art Modifier

The *Line Art* modifier generates stylized line art from a scene, collection, or object.

In order for the effects of the modifier to be visible, the scene must have an active camera. The generated lines are only generated on portions of the object that a visible from this camera.

Note

Due to lack of global cache at the moment, each Line Art modifier will run the entire occlusion calculation for itself. So if you have multiple line art modifiers to select different parts of the scene (to apply different styles, etc.), the evaluation will take much longer. There are plans to remedy this in the future, but this is a known limitation for now.

Options

Use Cache

Optimize rendering by using cached scene data from the first line art modifier in the stack. This option has the disadvantage of certain settings becoming unavailable.

This option is only available when you have more than one Line Art modifier in the same modifier stack and the modifier is not the first Line Art modifier in the stack.

Source Type

What type of geometry source should line art be generated from.

Scene, Collection, Object

Object/Collection

Based on the source type, collection or object can be selected as source geometry.

Invert Collection Filtering

Select everything except lines from specified collection.

Note

Line Art will still load and calculate the entire visible scene to produce correct occlusion result, unless specified to do otherwise in object or collection Line Art *Usage* property.

Layer

The Grease Pencil [Layers](#) to put the result in.

Material

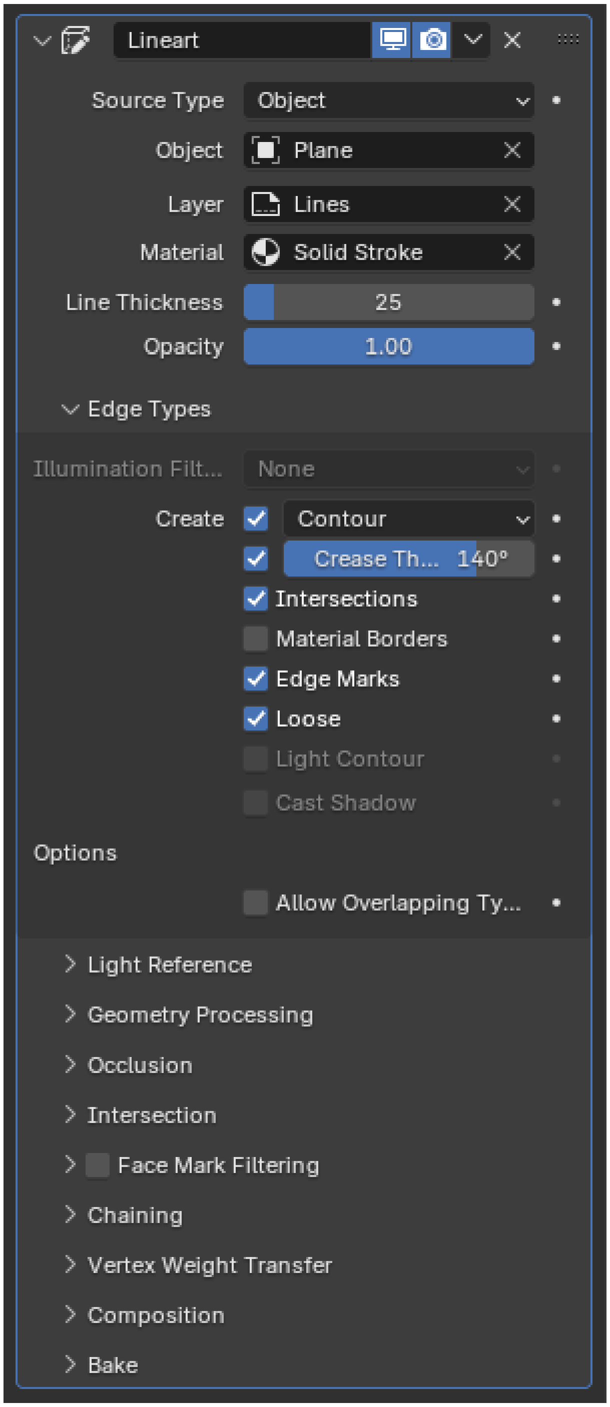
The Grease Pencil [Grease Pencil Materials](#) to generate strokes with.

Line Thickness

The strokes generated by line art are given this thickness.

Opacity

The strokes generated by line art are given this Opacity.



Lineart

Source Type: Object

Object: Plane

Layer: Lines

Material: Solid Stroke

Line Thickness: 25

Opacity: 1.00

Edge Types

Illumination Filt...: None

Create: ☑ Contour

☑ Crease Th... 140°

☑ Intersections

☐ Material Borders

☑ Edge Marks

☑ Loose

☐ Light Contour

☐ Cast Shadow

Options

☐ Allow Overlapping Ty...

Light Reference
Geometry Processing
Occlusion
Intersection
☐ Face Mark Filtering
Chaining
Vertex Weight Transfer
Composition
Bake

The Line Art modifier.

Edge Types

Line Art can identify different edge types. Selected edge types will be included in the result.

Illumination Filtering

Select feature lines that comes from lit or shaded regions. Will not affect cast shadow and light contour since they are at the border.

None::	Not filtering any lines based on illumination region.
Illuminated::	Only selecting lines from illuminated regions.
Shaded::	Only selecting lines from shaded regions.
Illuminated (Enclosed Shapes)::	Selecting lines from lit regions, and make the combination of contour, light contour and shadow lines into enclosed shapes.

Create

Contour

Generate strokes from contour lines. Where the edge becomes the separation line of front/backfacing faces. The silhouette can also be inverted by clicking the invert button.

Contour::	Generate lines from contour.
Silhouette::	Only generate lines from the silhouette of the source objects as a whole.
Individual Silhouette::	Generate lines from the individual silhouettes of the source objects.

Crease

Generate strokes where the edge angle is small enough.

Crease Threshold

Angles smaller than this will be treated as creases. Crease angle priority: object line art crease override > line art default crease.

Intersections
Generate strokes where lines intersect between faces.

Material Borders
Generate strokes where the edge separates faces with different materials.

Edge Marks
Generate strokes from freestyle edge marks.

Loose
Generate strokes for edges that do not form a Face.

Light Contour
Generate light/shadow separation lines from a reference Light Object.

Cast Shadow
Project contour lines using a light source object.

Options

Allow Overlapping Types
Allow an edge to have multiple overlapping types. This will create a separate stroke for each overlapping type.

Light Reference

Light Object
Use this light object to generate Light Contour.

Shadow Camera Size
This value represents the “Orthographic Scale” of an ortho camera. If the camera is put at the lamps position with this scale, it will represent the coverage of the shadow “camera”.

Near

Near clipping distance of shadow camera.

Far

Far clipping distance of shadow camera

Geometry Processing

Custom Camera

Use custom camera instead of the active camera for calculating strokes. Useful when baking multiple shots in different angles as well as for motion graphics effects.

Overlapping Edges as Contour

This option allows overlapping edges (e.g. from an Edge Split modifier or imported geometry where two edges occupy the exact same space) to be drawn as contour. Enabling this option will slow down the calculation slightly but it will handle edge overlapping cases without erroneous occlusion results.

Instanced Objects

This option enables particles and other instanced objects to be loaded for line art calculation. There will be performance impact when there are a large amount of instanced objects in the scene.

Clipping Boundaries

When enabled, line art will generate clipping lines as contour type at the place where near or far clipping planes cut the model. Otherwise there will be no lines.

Crease on Smooth

Allow crease edges to show inside smooth surfaces.

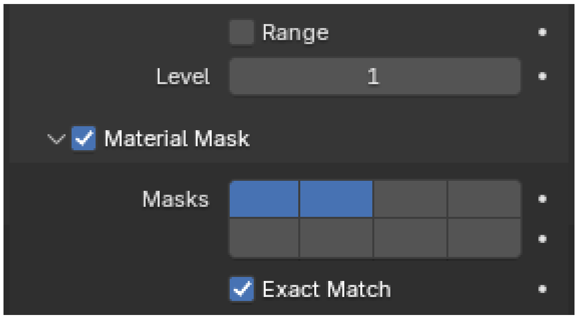
Crease on Sharp

Allow creases to show on sharp edges.

Force Backface Culling

Remove all back faces to speed up calculation. Note, removing back faces will create edges in different occlusion levels than when disabled.

Occlusion


☐ Range

Level: 1

☑ Material Mask

☑ Exact Match

Occlusion subpanel.

Range

If enabled, the modifier will select lines that have an occlusion level between start and end values.

Level

Desired occlusion level to be selected as line art result. A value of 0 means visible lines (no occlusion). A value of 1 means selecting lines that have been occluded by exactly one layer of faces.

Material Mask

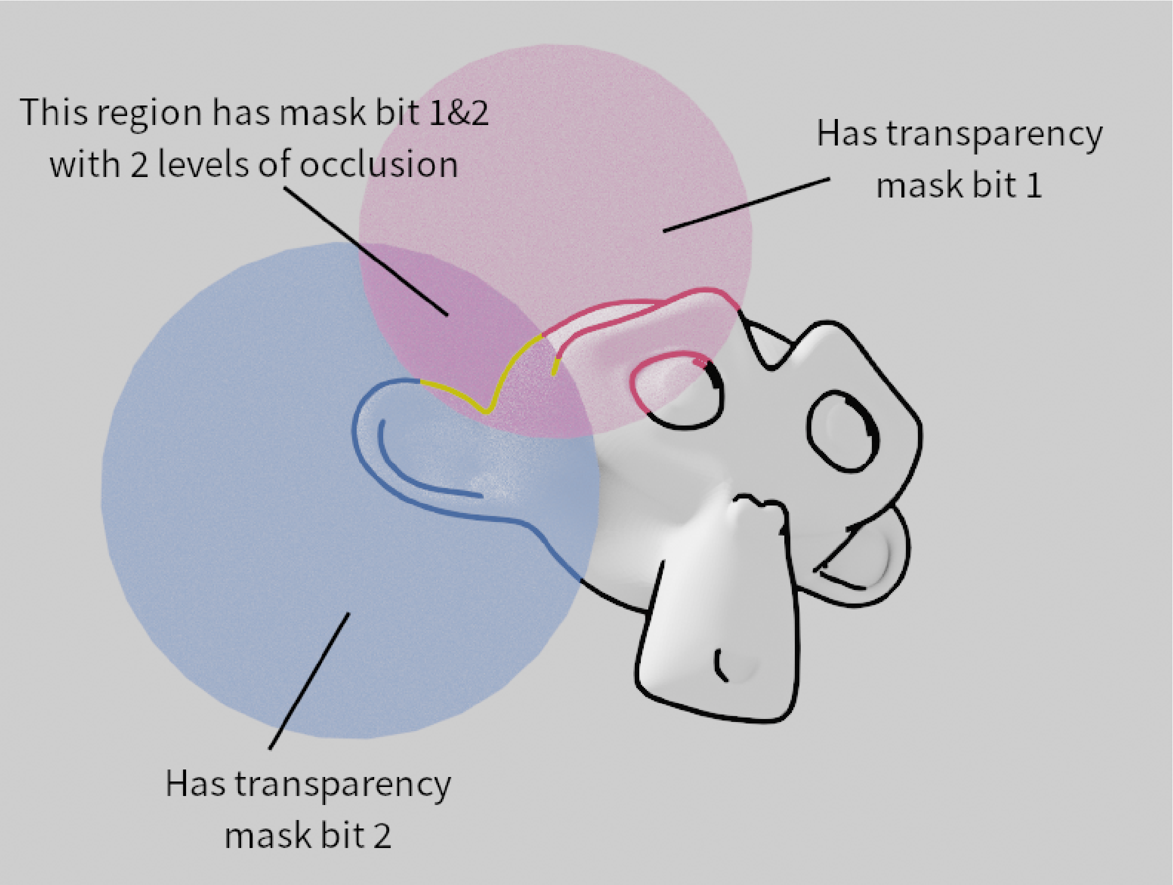
If enabled, Line Art will only select lines that are occluded by certain faces whose material have specific occlusion masks set.

Masks

To select edges that have been occluded by the selected Material Mask.

Exact Match

If enabled, only lines that are occluded with the exact mask bit combination will be selected. Otherwise, lines that have been occluded by any one of specified material masks will be selected.


This region has mask bit 1&2
with 2 levels of occlusion
Has transparency
mask bit 1
Has transparency
mask bit 2

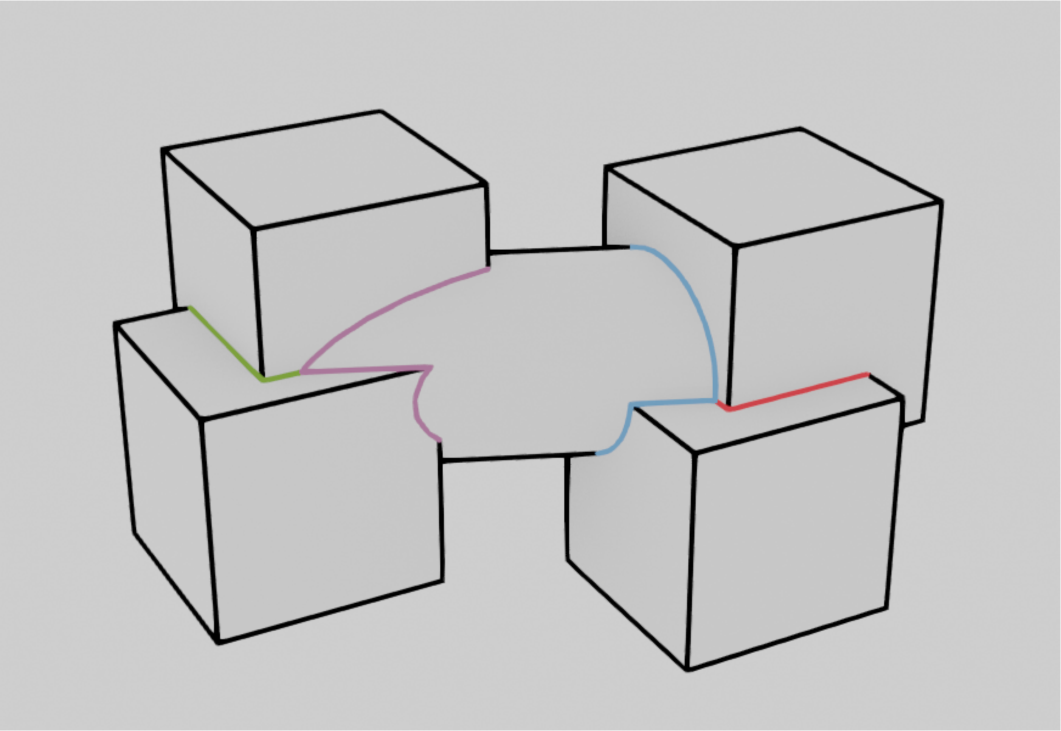
Demonstration of the usage of material masks.

Intersection

Allows you to select edges that intersect between two collections.

Collection Mask
: Mask bits to match from [Collection Line Art](#) properties.

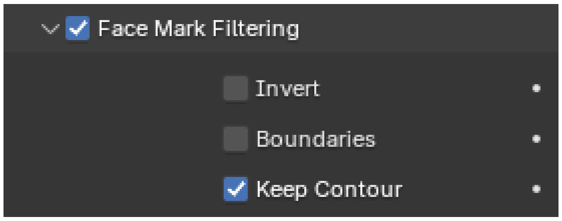
Exact Match
: Require matching all intersection masks instead of just one.



Demonstration of the usage of collection masks.

Face Mark Filtering

Face Mark Filtering can be used to have manual control over which feature edges produce strokes by using Freestyle face marks.



☑ Face Mark Filtering

☐ Invert

☐ Boundaries

☑ Keep Contour

Face Mark Filtering subpanel.

Invert

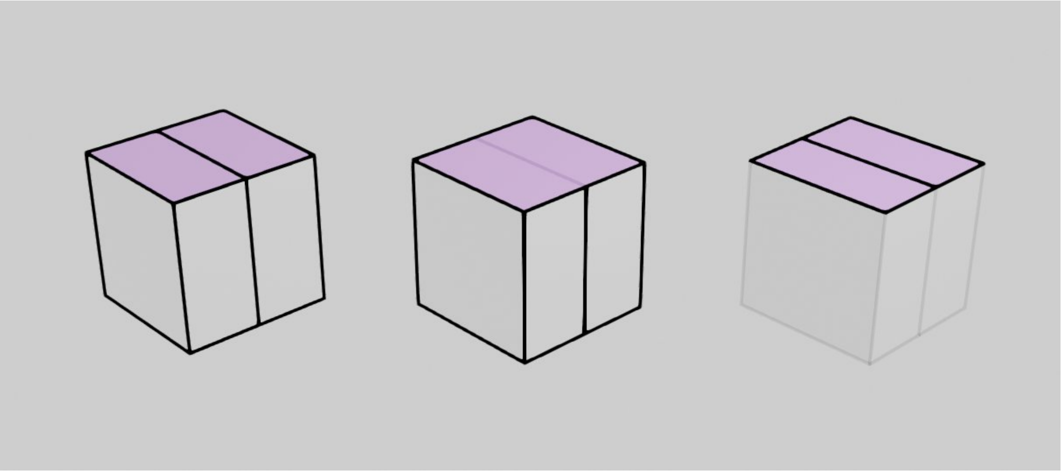
Invert face mark filtering.

Boundaries

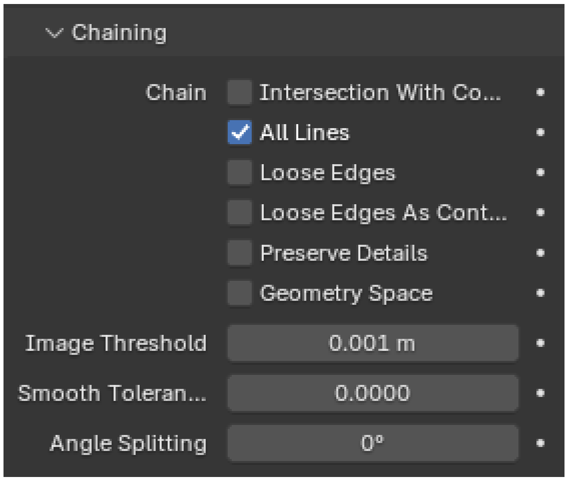
Filter feature lines based on face mark boundaries.

Keep Contour

Preserve contour lines while filtering.



Chaining



Chaining

Chain ☐ Intersection With Co...

☑ All Lines

☐ Loose Edges

☐ Loose Edges As Cont...

☐ Preserve Details

☐ Geometry Space

Image Threshold 0.001 m

Smooth Toleran... 0.0000

Angle Splitting 0°

Chaining subpanel.

Chain

Intersection with Contour

Allows intersection lines to be chained together with contour lines.

Note

Enabling this option will lead to ambiguity in intersection edge types. Intersection lines that have not been able to chain with any nearby contour lines will remain as intersection lines.

All Lines

Enabling this option will cause all lines to have the type of contour and to be chained together.

Loose Edges

Allow floating Edges that do not form a face to be chained together.

Loose Edges as Contour

Edges that do not form a face will be classified as contour lines.

Preserve Details

Instead of splitting at each occlusion change, keep small details from the initial chain. When details are not kept, will create a much smoother result.

Geometry Space

Use geometry distance for chaining instead of image space.

Image Threshold

Allow the end point of short segments to be chained together if the 2D image space distance between them are within the specified threshold.

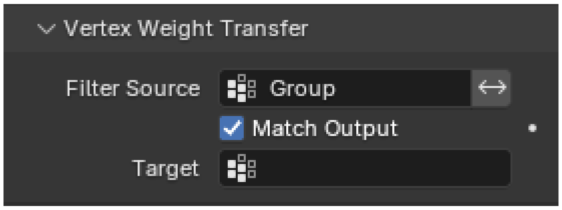
Smooth Tolerance

The strength of smoothing applied on jagged chains.

Angle Splitting

Split a chain at sharp “turning” points specified by this angle.

Vertex Weight Transfer



Filter Source: Group

☑ Match Output

Target:

Vertex Weight Transfer subpanel.

Filter Source

If source mesh has vertex groups whose name starts with this text, then the vertex weight info will be transferred into weight groups in Grease Pencil strokes.

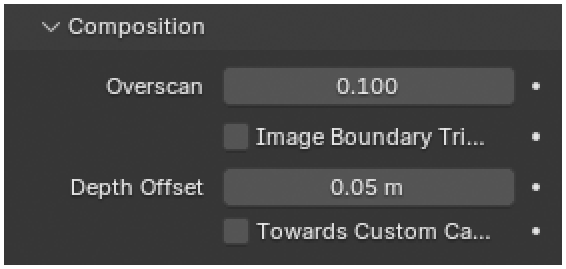
Match Output

Transfer the filtered object vertex weights into Grease Pencil weight groups with the same names as the filtered ones.

Target

If *Match Output* is off, then a target vertex group has to be specified. If there are multiple weight groups copied into target, then the highest weight value is copied into it.

Composition



Overscan 0.100

☐ Image Boundary Tri...

Depth Offset 0.05 m

☐ Towards Custom Ca...

Composition subpanel.

Overscan

To optimize rendering, Blender only renders the strokes for edges of the object that are in the camera’s view. This optimization however, can result in strokes ending abruptly at the edge of the image.

This value prevents this error by adding a margin outside the camera’s view to continue computing strokes.

Image Boundary Trimming

Trim all stroke right at the boundary of image (including overscan region).

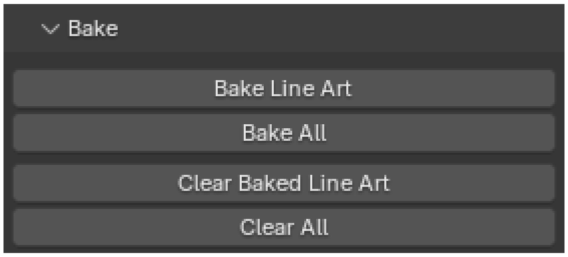
Depth Offset

Move strokes slightly towards the camera to avoid clipping while preserve depth for the viewport. This option is unavailable unless Show in Front is disabled.

Towards Custom Camera

Offset strokes towards selected camera (see *Custom Camera* above) instead of the active camera.

Bake



Bake Line Art

Bakes Line Art strokes for active Grease Pencil object within the *start*, *end* frame range in scene. After baking, baked Line Art modifiers will be deactivated automatically.

Bake options.

Bake All

Bakes all Grease Pencil objects that contains at least one Line Art modifier. After baking, baked Line Art modifiers will be deactivated automatically.

Clear Baked Line Art

Clears baked line art frames within the scene frame range for active Grease Pencil object.

Clear All

Clears baked line art for all Grease Pencil objects that contains at least one Line Art modifier.

Warning

If you have drawn anything manually in the frame range of where line art runs, this operation will also clear those strokes!

Continue without Clearing

Re-activate a specific Line Art modifier without clearing baked strokes. This is useful for working on multiple portions of frames separately.

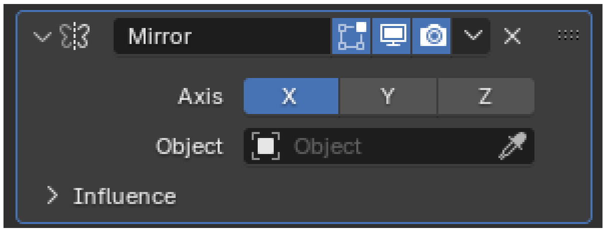
Mirror Modifier

The *Mirror* modifier mirrors the strokes along its local X, Y and/or Z axes, across the Object Origin. It can also use another object as the mirror center, then use that object's local axes instead of its own.

See also

This documentation refers to the Mirror Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Mirror Modifier.

Options



The Mirror modifier.

Axis

The X, Y, Z axis along which to mirror, i.e. the axis perpendicular to the mirror plane of symmetry.

To understand how the axis applies to the mirror direction, if you were to mirror on the X axis, the positive X values of the original stroke would become the negative X values on the mirrored side.

You can select more than one of these axes. And will then get more mirrored copies. With one axis you get a single mirror, with two axes four mirrors, and with all three axes eight mirrors.

Object

A Data ID to select an object (usually an empty), which position and rotation will be used to define mirror planes (instead of using the ones from the modified object).

Influence

See Influence Filters.

Multiple Strokes

The *Multiple Strokes* modifier generate multiple parallel strokes around the original ones.

Options

Duplicates

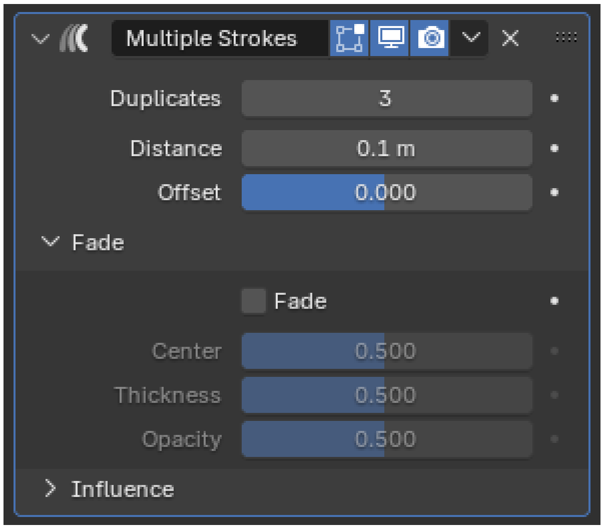
The number of additional strokes.

Distance

Distance between the original and the duplicate strokes.

Offset

Control the offset position (inner or outer) for duplicate strokes.



Multiple Strokes

Duplicates 3

Distance 0.1 m

Offset 0.000

Fade

☐ Fade

Center 0.500

Thickness 0.500

Opacity 0.500

Influence

The Multiple Strokes modifier.

Fade

When activated, the duplicate strokes fades out using their opacity or thickness.

Center

Control the initial position for the fading.

Thickness

Fade influence on strokes thickness.

Opacity

Fade influence on strokes opacity.

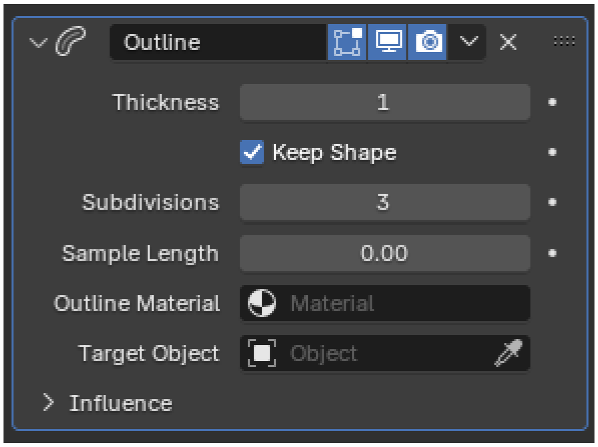
Influence

See Influence Filters.

Outline Modifier

The *Outline* modifier convert strokes to outline tracing all strokes perimeter with new strokes.

Options


Outline

Thickness: 1

☑ Keep Shape

Subdivisions: 3

Sample Length: 0.00

Outline Material: Material

Target Object: Object

Influence

The Outline modifier.

Thickness

The thickness of the generated strokes outline.

Keep shape

The perimeter strokes are maintaining inside the original stroke perimeter trying to keep the original shape.

Subdivisions

controls the number of subdivision of the generated strokes outline.

Sample Length

Controls the accuracy of the perimeter conversion.

Outline Material

Defines the material to use on the generated strokes outline.

Target Object

Controls the origin of the cyclic strokes generated.

Influence Filters

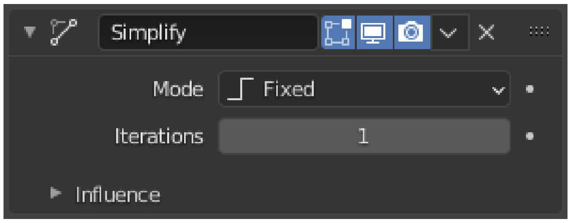
See Influence Filters.

Simplify Modifier

The *Simplify* modifier allows you to reduce the amount of points in the strokes. The goal of this modifier is reduce points while maintaining the lines shape.

Apply the modifier can help to obtain a better performance (more FPS) while animating.

Options


Simplify

Mode: Fixed

Iterations: 1

The Simplify modifier.

Mode
Determines how to reduce points in the strokes.

Fixed::	Deletes alternated points in the strokes, except the start and end points. Iterations Number of times to repeat the procedure.
Adaptive::	Uses the RDP algorithm (Ramer-Douglas-Peucker algorithm) for points deletion. The algorithm try to obtain a similar line shape with fewer points. Factor Controls the amount of recursively simplifications applied by the algorithm.
Sample::	Recreates the stroke geometry with a predefined length between points. Length The distance between points on the recreated stroke. Smaller values will require more points to

recreate the stroke, while larger values will result in fewer points needed to recreate the curve.

Sharp Threshold

Preserve corners that have sharper angle than this threshold.

Merge:: Simplifies the strokes by merging points that are closer than a specified distance to each other.

Distance

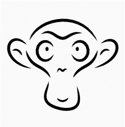
Sets the distance threshold for merging points.

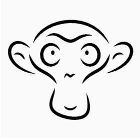
Influence

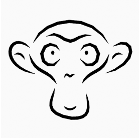
See Influence Filters.

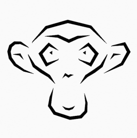
Example

Fixed Mode sample.

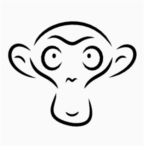

Original Model.

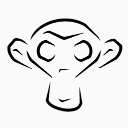

Iteration: 1.

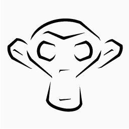

Iteration: 2.

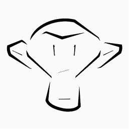

Iteration: 3.

Adaptive Mode sample.


Original Model.


Factor: 0.1.


Factor: 0.5.


Factor: 1.

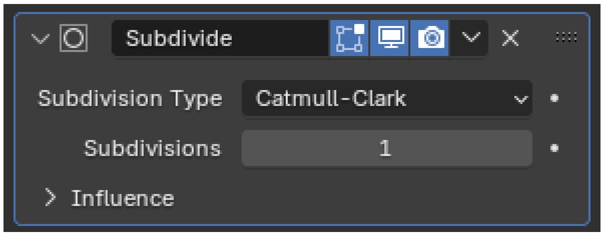
Subdivide Modifier

The *Subdivide* modifier subdivide the strokes by inserting points between other points to the lines.

Options

Subdivision Type

Catmull-Clark::



Subdivide

Subdivision Type: Catmull-Clark

Subdivisions: 1

The Subdivide modifier.

Subdivides and smooths the surfaces.

Simple:: Only subdivides the surfaces, without any smoothing.

Subdivisions

Recursively adds more points.

Influence

See Influence Filters.

Grease Pencil Deform Modifiers

- Armature Modifier
- Hook Modifier
- Lattice Modifier
- Noise Modifier
- Offset Modifier
- Shrinkwrap Modifier
- Smooth Modifier
- Thickness Modifier

Armature Modifier

The *Armature* Modifier is used for building skeletal systems for animating the poses of characters and anything else which needs to be posed.

By adding an armature to an object, this object can be deformed accurately so that geometry does not have to be animated by hand.

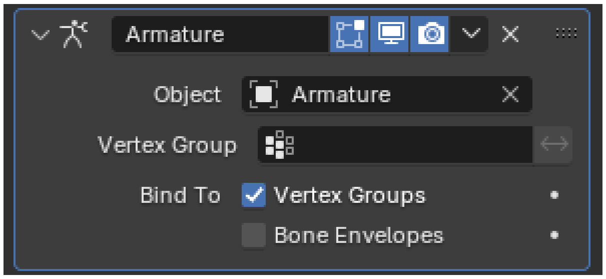
See also

For more details on armatures usage, see the armature section.

See also

This documentation refers to the Armature Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Armature Modifier.

Options



Object: Armature
Vertex Group:
Bind To: ☑ Vertex Groups
☐ Bone Envelopes

The Armature modifier.

Object
: The name of the armature object used by this modifier.

Vertex Group
: The name of a vertex group of the object, the weights of which will be used to determine the influence of this Armature Modifier's result when mixing it with the results from other *Armature* ones.

Only meaningful when having at least two of these modifiers on the same object, with *Multi Modifier* activated.

Invert <–>

Inverts the influence set by the vertex group defined in previous setting (i.e. reverses the weight values of this group).

Bind to

Vertex Groups

When enabled, bones of a given name will deform points which belong to vertex groups of the same name. E.g. a bone named “forearm”, will only affect the points in the “forearm” vertex group.

The influence of one bone on a given point is controlled by the weight of this point in the relevant group. A much more precise method than *Bone Envelopes*, but also generally longer to set up.

Bone Envelopes

When enabled, bones will deform points or control points near them, defined by each bone’s envelope radius and distance. Enable/Disable bone envelopes defining the deformation (i.e. bones deform points in their neighborhood).

Hook Modifier

The *Hook* Modifier is used to deform stroke points using another object (usually an empty or a bone but it can be any object).

As the hook moves, it pulls points from the strokes with it. You can think of it as animated Proportional Editing.

See also

This documentation refers to the Hook Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Hook Modifier.

Options

Object

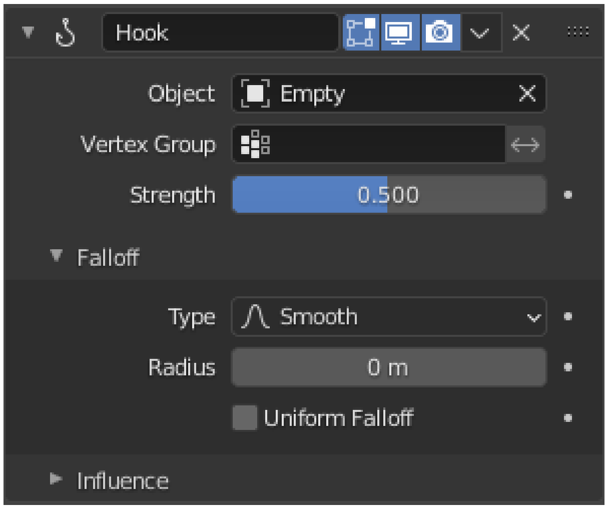
The name of the object to hook points to.

Vertex Group

Restricts the effect only to a vertex group.

Strength

Adjust this hooks influence on the stroke points, were (0.0 to 1.0) (no change to fully follow the hook).



Object: Empty

Vertex Group:

Strength: 0.500

Type: Smooth

Radius: 0 m

☐ Uniform Falloff

The Hook modifier.

Falloff

Type
: This can be used to adjust the type of curve for the Strength attenuation. You can also define a custom curve to get a much higher level of control.

Radius
: The size of the hooks influence.

Uniform Falloff
: This setting is useful when using hooks on scaled objects, especially in cases where non-uniform scale would stretch the result of the hook.

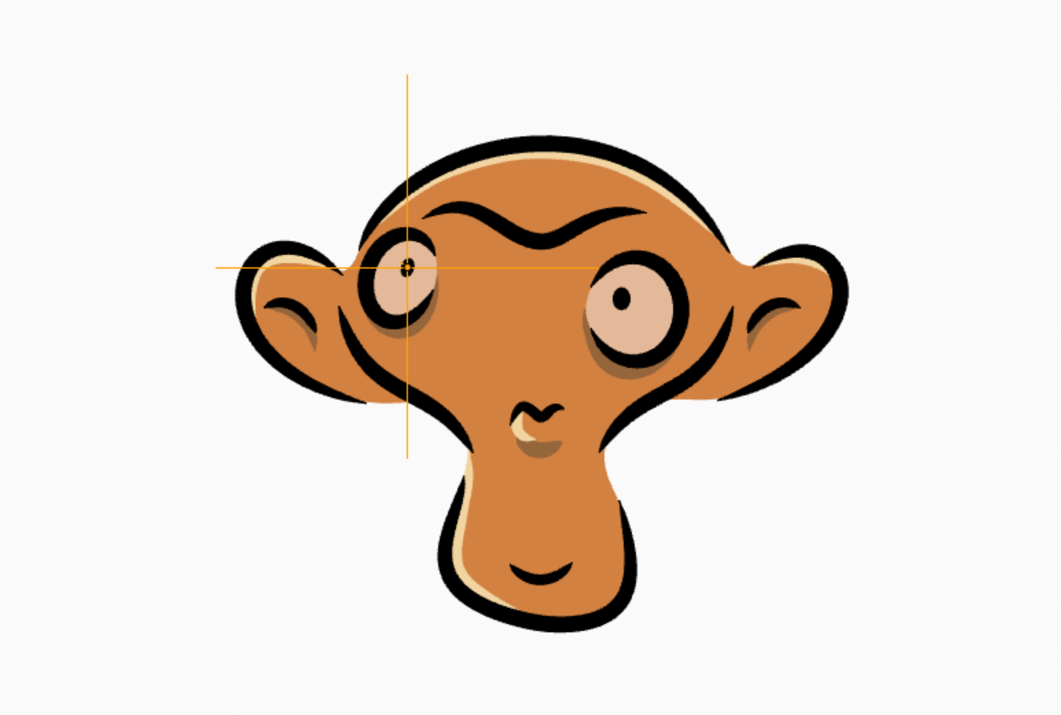
Influence

See Influence Filters.

Note

The Hook Modifier stores points indices from the original strokes to determine what to affect; this means that modifiers that generate geometry, like a Subdivision Surface Modifier, should always be applied **after** the Hook Modifier; otherwise the generated geometry will be left out of the hook's influence.

Example



Empty used as a hook to manipulate a vertex group (right eye of the monkey).

Lattice Modifier

The Lattice modifier deforms the base object according to the shape of a Lattice object.

Tip

A Lattice Modifier can quickly be added to selected objects by selecting them all, then selecting the lattice object last and pressing `Ctrl-P` and choosing *Lattice Deform*. This will both add Lattice Modifiers to the selected objects and parent them to the lattice.

See also

This documentation refers to the Lattice Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Lattice Modifier.

Options

Object

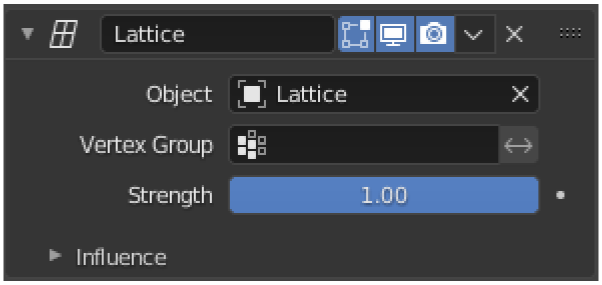
The Lattice object with which to deform the base object.

Vertex Group

Restricts the effect only to a vertex group.

Strength

A factor to control blending between original and deformed points positions.



Object: Lattice
Vertex Group
Strength: 1.00

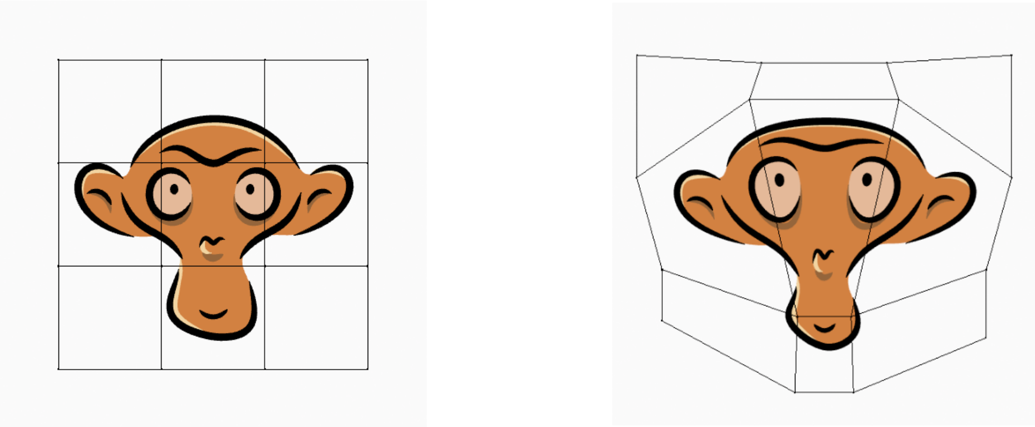
Lattice Modifier.

Influence

See Influence Filters.

Example

Lattice modifier example.



Original model.

After lattice edition.

Noise Modifier

The *Noise* Modifier changes the value of one or more stroke/points properties like: location, strength, thickness or UV texture position by adding varied values that make the line unstable and noisy.

Random values can be used for the noise factor for more vivid effects.

Options

Position

Strength of the noise effect over the point location.

Strength

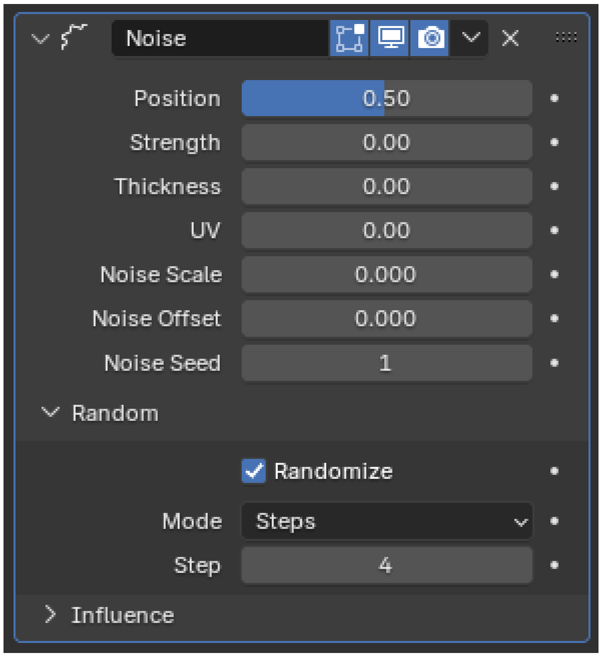
Strength of the noise effect over the point strength (opacity).

Thickness

Strength of the noise effect over the point thickness.

UV

Strength of the noise effect over the point UV rotation.



Position 0.50
Strength 0.00
Thickness 0.00
UV 0.00
Noise Scale 0.000
Noise Offset 0.000
Noise Seed 1
☑ Randomize
Mode Steps
Step 4

Noise Modifier.

Noise Scale

Control the noise frequency scale.

Noise Offset

Moves the noise along the strokes.

Seed

Seed used by the pseudo-random number generator.

Randomize

When enabled, the noise uses a random value over time.

Mode

Steps::	New random value at defined steps. Step Number of frames before using a new random value.
Keyframes::	New random value only on keyframes.

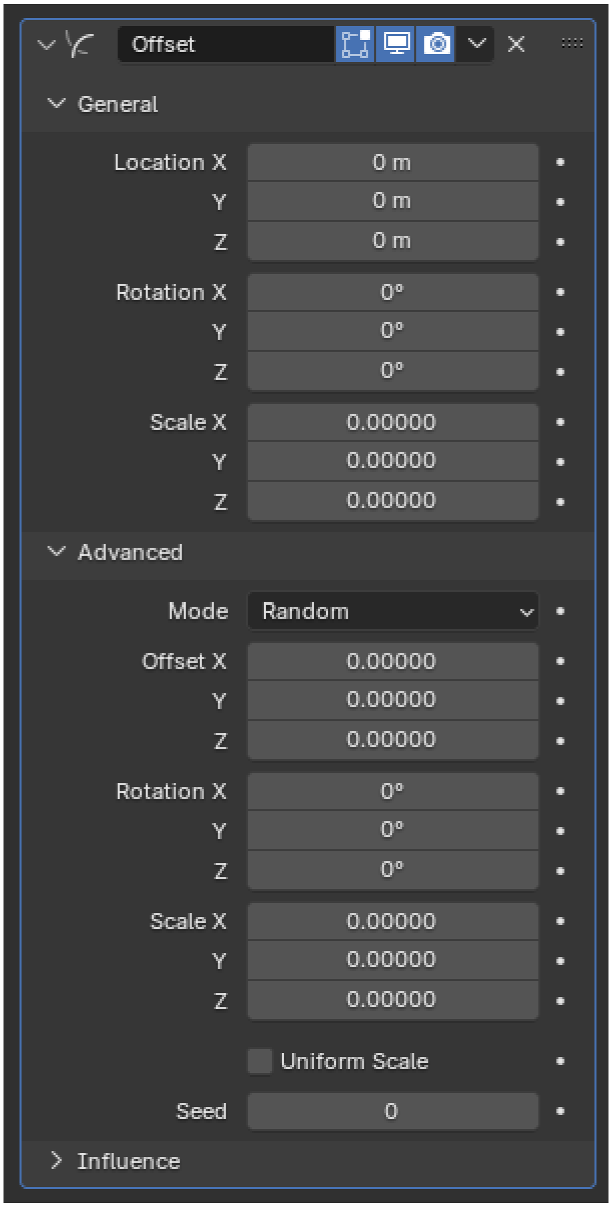
Influence

See Influence Filters.

Offset Modifier

The *Offset* Modifier changes the strokes location, rotation or scale starting from the object origin.

Options



General

Location X 0 m
Y 0 m
Z 0 m

Rotation X 0°
Y 0°
Z 0°

Scale X 0.00000
Y 0.00000
Z 0.00000

Advanced

Mode Random

Offset X 0.00000
Y 0.00000
Z 0.00000

Rotation X 0°
Y 0°
Z 0°

Scale X 0.00000
Y 0.00000
Z 0.00000

☐ Uniform Scale

Seed 0

Influence

Offset Modifier.

General

Location X, Y, Z
Sets strokes location offset from its object origin.

Rotation X, Y, Z
Sets strokes rotation.

Scale X, Y, Z
Sets strokes scale.

Advanced

Mode

Random::	Add random values to the individual strokes offset.
Layer::	Offset by layers.
Stroke::	Offset by strokes (based on the stroke draw order).
Material::	Offset by Materials.

Offset X, Y, Z
Sets individual element location offset.

Rotation X, Y, Z
Sets individual element rotation.

Scale X, Y, Z
Sets individual element scale.

Uniform Scale (Random mode)
Use the same random *Seed* for each scale axis in the strokes for a uniform scale.

Seed (Random mode)

Seed used by the pseudo-random number generator.

Layer/Stroke/Material Step (For Layer, Stroke and Material mode)
The number of elements to be grouped and offset together.

Offset (For Layer, Stroke and Material mode)
Offset the element starting point.

Influence

See Influence Filters.

Shrinkwrap Modifier

The *Shrinkwrap* modifier allows a Grease Pencil object to “shrink” to the surface of another object. It moves each point of the object being modified to the closest position on the surface of the given mesh.

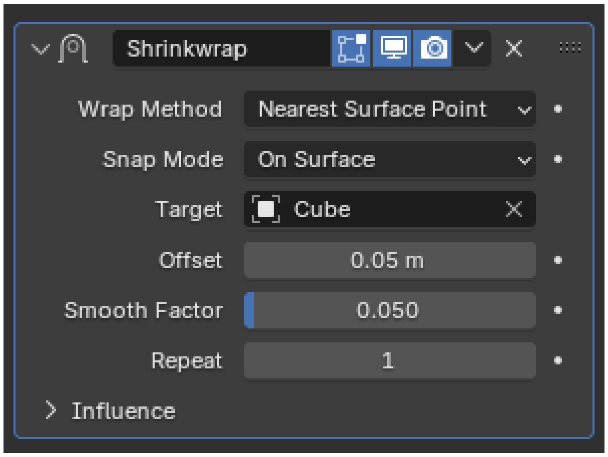
See also

Shrinkwrap Constraint.

See also

This documentation refers to the Shrinkwrap Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Shrinkwrap Modifier.

Options



Wrap Method: Nearest Surface Point
Snap Mode: On Surface
Target: Cube
Offset: 0.05 m
Smooth Factor: 0.050
Repeat: 1

The Shrinkwrap modifier in Nearest Surface Point mode.

Wrap Method

This selector specifies the method to be used to determine the nearest point on the target’s surface for each point of the modified object. Some options will add some extra, specific controls to the panel. See Wrap Methods for an explanation of each method.

Snap Mode

Most modes support an additional setting to control how the point is moved to the target point selected by the methods described above. Some of the choices only differ if *Offset* is not zero.

On Surface::	The point is always moved. The offset is applied along the projection line connecting the original point and selected target point towards the original position.
Inside::	The point is not moved if it is already inside the target. Offset shrinks the allowed volume towards the inside along the projection line.
Outside::	The point is not moved if it is already outside the target. Offset expands the exclusion volume towards the outside along the projection line.
Outside Surface::	Like *On Surface*, but the offset is always applied towards the outside of the target.
Above Surface::	Like *On Surface*, but the offset is applied along the smooth normal of the target.

Note

The *Inside* and *Outside* options can be used for very crude collision detection. The inside vs outside determination is done based on the target normal and is not always stable near 90 degree and sharper angles in the target mesh.

Target

Shrink target, the mesh to shrink to/wrap around.

Offset

The distance that must be kept from the calculated target position.

Smooth Factor

Amount of smoothing to apply.

Repeat
The number of time to apply smoothing.

Influence

See Influence Filters.

Wrap Methods

Nearest Surface Point

This will select the nearest point over the surface of the shrunk target.

Project

This will project vertices along a chosen axis until they touch the shrink target. Vertices that never touch the shrink target are left in their original position.

Limit
This is a distance limit between original point and surface. If the distance is larger than this limit point would not be projected onto the surface.

Subdivision Levels
This applies a (temporary) *Catmull-Clark* subdivision to the modified object's geometry, before computing the wrap.

Axis
Along which local axis of the modified object the projection is done. These options can be combined with each other, yielding a "median axis" of projection. If none are selected, the normal direction is used.

Negative/Positive
This allows you to select the allowed direction(s) of the shrink along the selected axis. If both options are enabled, both ways are evaluated and the closest hit is selected.

Face Cull

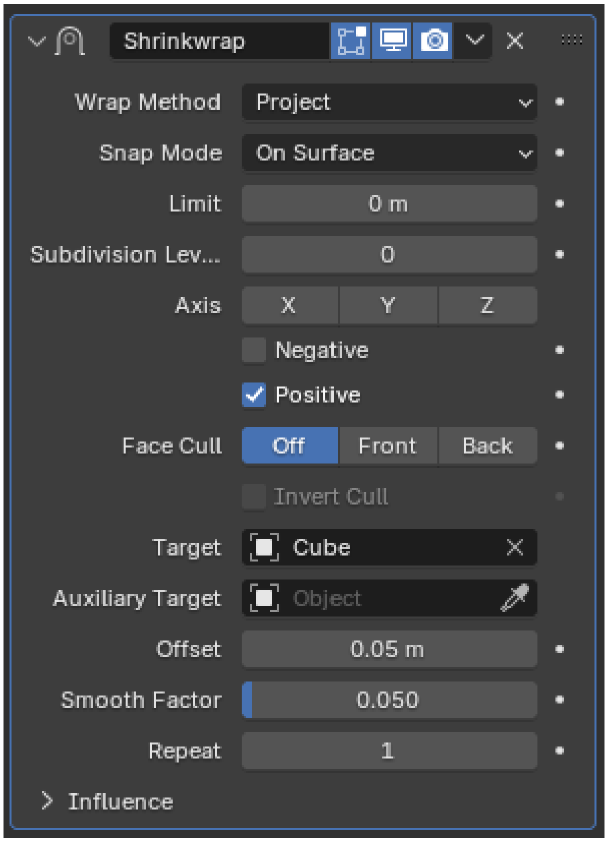
Allows you to prevent any projection over the "front side" (respectively the "back side") of the target's faces. The "side" of a face is determined by its normal (front being the side "from where" the normal "originates").

Invert Cull

If *Cull Faces* is enabled, and *Negative* direction along axis is allowed, this option can be used to invert the *Front* or *Back* cull choice for the *Negative* direction. This is useful when projecting in both directions.

Auxiliary Target

An additional object to project over.



Shrinkwrap

Wrap Method: Project

Snap Mode: On Surface

Limit: 0 m

Subdivision Lev...: 0

Axis: X Y Z

☐ Negative

☑ Positive

Face Cull: Off Front Back

☐ Invert Cull

Target: Cube

Auxiliary Target: Object

Offset: 0.05 m

Smooth Factor: 0.050

Repeat: 1

Influence

Project mode.

Nearest Vertex

This will snap vertices to the nearest vertex of the shrunk target. It adds no extra options.

This method doesn't support the *Snap Mode* setting.

Target Normal Project

This mode is similar to *Nearest Surface Point*, but produces a much smoother projection in return for being significantly slower.

Instead of finding the closest point, it searches for the nearest point that has its interpolated smooth normal pointing towards or away from the original point position. Non-manifold boundary edges are specially handled as infinitely thin cylinders that emit normals in all perpendicular directions; ignores flat shading.

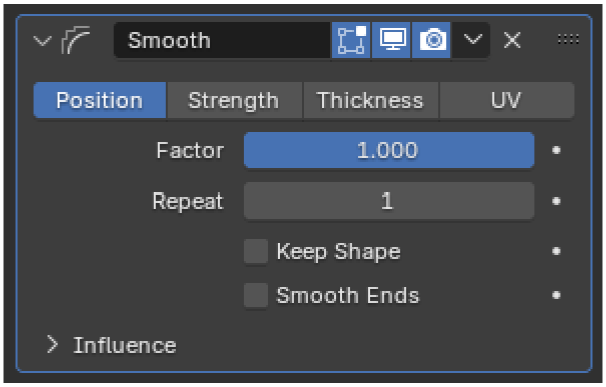
Smooth Modifier

The *Smooth* Modifier changes the value of one or more stroke/points properties like: location, strength, thickness or UV texture position trying to maintain similar values that make the line fluid and smoother.

Options

Affect

Combination of stroke/points properties that will be affected by the smooth factor.



Factor 1.000

Repeat 1

☐ Keep Shape

☐ Smooth Ends

Smooth Modifier.

Position::	Smooth affect the point location.
Strength::	Smooth affect the point strength (opacity).
Thickness::	Smooth affect the point thickness.
UV::	Smooth affect the point UV rotation.

Factor

Strength of the smooth effect.

Repeat

The number of smoothing iterations, equivalent to executing the Smooth tool multiple times. High values can reduce the animation performance (FPS).

Keep Shape

When enabled, the smoothing algorithm will try to maintain as close as possible the overall shape of the original stroke.

Smooth Ends
Smooth the ends of the strokes.

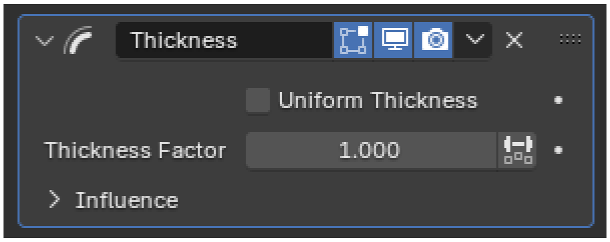
Influence

See Influence Filters.

Thickness Modifier

The *Thickness* Modifier change the stroke points thickness.

Options



☐ Uniform Thickness

Thickness Factor 1.000

Thickness Modifier.

Uniform Thickness
  When enabled, makes the thickness equal for the entire strokes.

  Thickness
    Absolute Thickness for the stroke points.

Thickness Factor
  Value to add or subtract to the actual points thickness.

Influence

See Influence Filters.

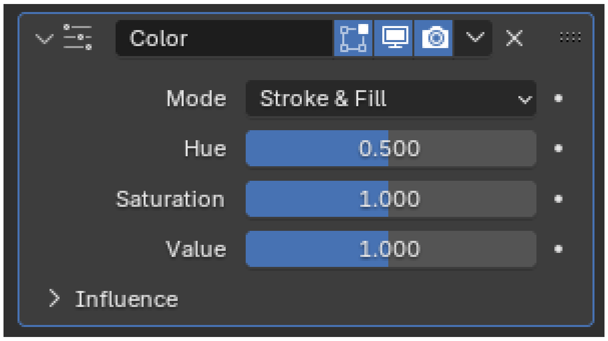
Grease Pencil Color Modifiers

- Hue/Saturation Modifier
- Opacity Modifier
- Tint Modifier

Hue/Saturation Modifier

The *Hue/Saturation* Modifier applies a color transformation to the object output color.

Options


Color

Mode: Stroke & Fill

Hue: 0.500

Saturation: 1.000

Value: 1.000

Hue/Saturation Modifier.

Mode

> The color transformation will be applied on the stroke and/or the fill color.
>
> Stroke & Fill, Stroke, Fill

Hue

> Specifies the hue rotation of the image. 360° are mapped to (0 to 1). The hue shifts of 0 (-180°) and 1 (+180°) have the same result.

Saturation

> A saturation of 0 removes hues from the image, resulting in a grayscale image. A shift greater than 1.0 increases saturation.

Value

> Value is the overall brightness of the image. De/Increasing values shift an image darker/lighter.

Influence

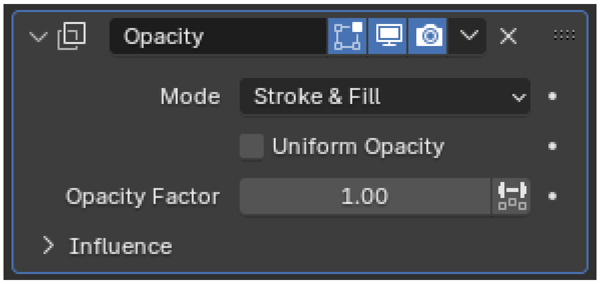
See Influence Filters.

Opacity Modifier

The *Opacity* Modifier change the opacity (alpha) value of the stroke points.

The alpha value in Grease Pencil is stored per-point. The modifier can alter these values to go from totally transparent points to totally opaque points.

Options



Mode: Stroke & Fill

☐ Uniform Opacity

Opacity Factor: 1.00

Opacity Modifier.

Mode

The color transformation will be applied to the stroke/fill color or stroke Hardness. When Hardness is selected the opacity affects the stroke’s transparency (alpha) from the center to the border.

Stroke & Fill, Stroke, Fill, or Hardness

Uniform Opacity

When enabled, makes the opacity equal for the entire strokes.

Strength

Absolute opacity for the stroke points.

Opacity Factor

Controls the opacity value of the stroke points. A value of 1.0 respect the original alpha value of the points, a shift less than 1.0 make the points more transparent than originally, and a shift greater than 1.0 make the points more opaque than originally.

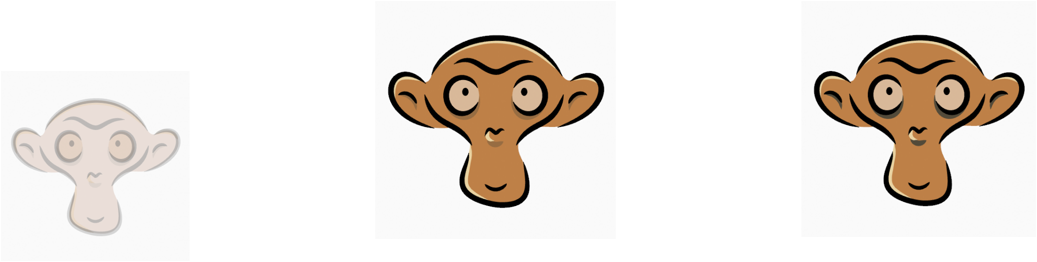
Sets value to 2.0 makes the points alpha fully opaque.

Influence

See Influence Filters.

Example

Opacity Factor samples.



Opacity Factor: 0.3.

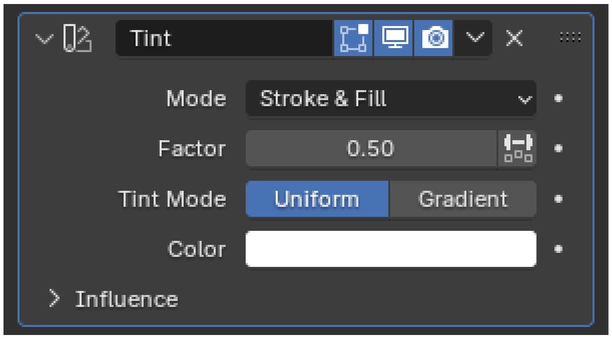
Opacity Factor: 1.0 (original alpha).

Opacity Factor: 2.0 (fully opaque).

Tint Modifier

The *Tint* Modifier colorize the original stroke or fill with a selected color.

Options



Mode Stroke & Fill

Factor 0.50

Tint Mode Uniform Gradient

Color

Tint Modifier.

Mode

The color transformation will be applied on the stroke and/or the fill color.

Stroke & Fill, Stroke, Fill

Strength

Controls the amount for the color mixing.

A value of 0 respect the original stroke's color, a value of 1.0 totally replace the original color with the tint color.

A shift greater than 1.0 will make the points alpha less transparent than originally (2.0 is fully opaque).

Tint Type

Uniform::

Color

Defines the tint color for mixing with the original color.

Gradient::

Color Ramp

Defines the tint gradient color for mixing with the original color. For controls see Color Ramp Widget.

Object

A Data ID to select an object (usually an empty), which position and rotation will be used to define the center of the effect.

Radius

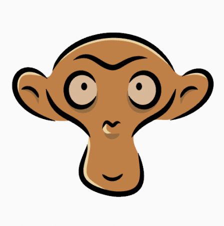
Defines the maximum distance of the effect.

Influence

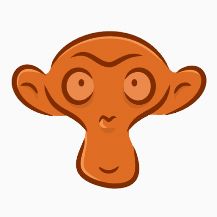
See Influence Filters.

Example

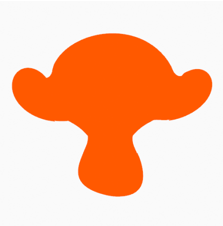
Tint uniform color sample.



Strength: 0 (original color).

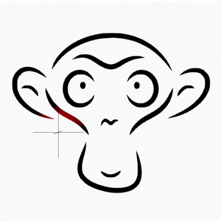


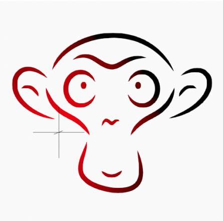
Strength: 0.5.

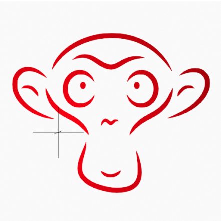


Strength: 1.0 (fully tinted).

Tint gradient color sample.







Radius: 1,
Strength: 1.

Radius: 5,
Strength: 1.

Radius: 10,
Strength: 1.

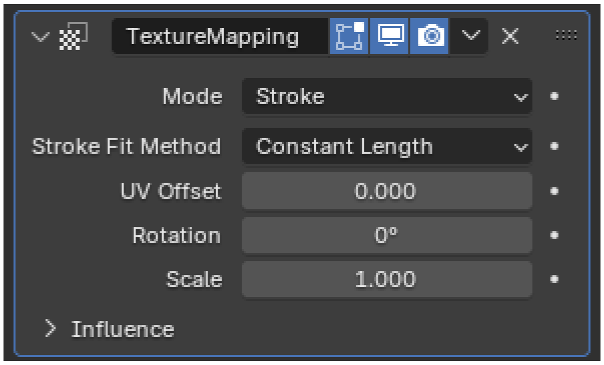
Grease Pencil Edit Modifiers

- Texture Mapping Modifier
- Time Offset Modifier
- Vertex Weight Angle Modifier
- Vertex Weight Proximity Modifier

Texture Mapping Modifier

The *Texture Mapping* Modifier change the strokes texture UV position.

Options



TextureMapping

Mode: Stroke

Stroke Fit Method: Constant Length

UV Offset: 0.000

Rotation: 0°

Scale: 1.000

Texture Mapping.

Mode

The texture transformation will be applied to the stroke/fill or stroke UVs.

Stroke::

Stroke Fit Method

Selects the texture fitting method.

Constant Length:: The texture keep a consistent length along the strokes.

Stroke Length:: The texture is normalized to fit the stroke length.

UV Offset

Moves the texture along the strokes.

Rotation

Rotates the points of the strokes.

Note
The *Rotation* option is limited to a range of -90 to 90 degrees.

Scale
    Factor for the texture scale.

Fill::

Fill Rotation
    Sets the texture angle.

Offset
    Moves the texture origin.
    X, Y

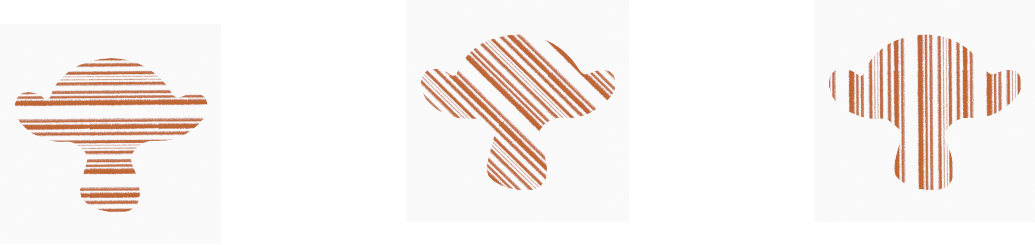
Scale
    Factor for the texture scale.

Influence

See Influence Filters.

Example

Opacity Factor samples.



Rotation: 0°.

Rotation: 45°.

Rotation: 90°.

Time Offset Modifier

The *Time Offset* Modifier applies a temporal offset to Grease Pencil keyframes on your timeline. If you have duplicated a Grease Pencil object you can use the Time Offset Modifier on the copies to desynchronize their animation. This can give more natural looking results.

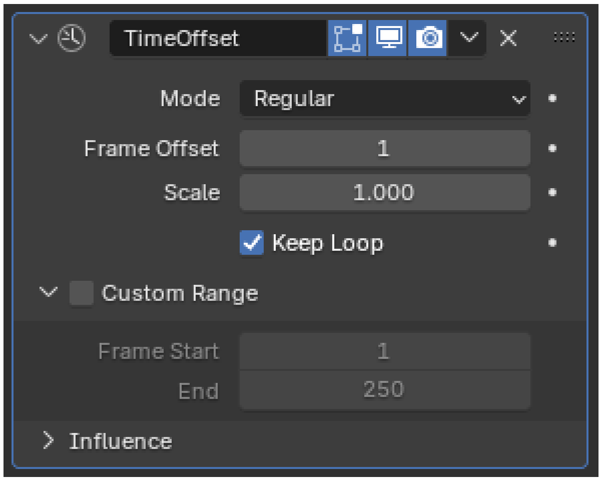
Using the Time Offset Modifier it's possible to have Grease Pencil frame ranges play back as repeating loops. Traditionally, 2D animation that uses looped drawings includes characters walking, rising smoke, and falling rain. In Fixed Frame mode the Time Offset Modifier can display drawings on your timeline entirely independently of the playhead position.

This can be handy for displaying drawings that will appear often in your animation. Think of switching between predefined mouth shapes for instance.

Options

Mode

Regular::



TimeOffset

Mode: Regular

Frame Offset: 1

Scale: 1.000

☑ Keep Loop

☐ Custom Range

Frame Start: 1

End: 250

Time Offset Modifier.

	Offsets keyframes in the default animation playback direction (playhead moving from left to right).
Reverse::	Offsets keyframes in reversed animation playback direction (playhead moving from right to left).
Fixed Frame::	The Frame parameter determines which frame is displayed. This value needs to be animated in order to have the displayed frame change during playback. Frame The number of the frame to display.
Ping Pong::	Loop back and forth animation.
Chain::	It allows to combine the different Modes consecutively. Repeat Number of cycle repeats

Frame Offset
The number of frames to offset the original keyframes by.

Scale
Controls the speed of the frames playback. 1 is equal to the actual frame rate, could be positive (faster) or negative (slower).

Keep Loop
Moves end frame to the animation start to keep animation in a loop.

Custom Range

When enabled, the animation playback is restricted only to a frame range.

Frame Start/End
Sets the range start and end frames.

Influence

See Influence Filters.

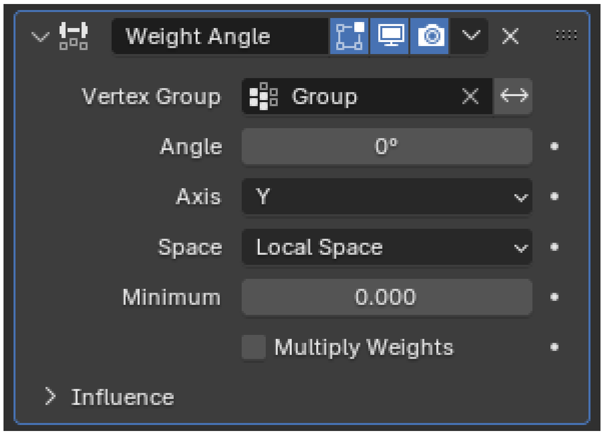
Vertex Weight Angle Modifier

This modifier sets the weights of the given vertex group, based on a predetermined angle.

Warning

This modifier does implicit clamping of weight values in the standard (0.0 to 1.0) range. All values below 0.0 will be set to 0.0, and all values above 1.0 will be set to 1.0.

Options



Weight Angle

Vertex Group: Group

Angle: 0°

Axis: Y

Space: Local Space

Minimum: 0.000

☐ Multiply Weights

The Vertex Weight Proximity modifier panel.

Vertex Group
  The vertex group to affect.

  Invert <-->
    Inverts the influence of the selected vertex group. The setting reverses the weight values of the group.

Angle

Sets the angle for the maximum weights value.

Axis

The axis along which the angle affects the weights.

X, Y, Z

Space

Coordinate space to be used.

Minimum

Minimum value for vertex weight.

Multiply Weights

Multiply the calculated weights with the existing values in the vertex group.

Influence

See Influence Filters.

Vertex Weight Proximity Modifier

This modifier sets the weights of the given vertex group, based on the distance between the object (or its vertices), and another target object (or its geometry).

Warning

This modifier does implicit clamping of weight values in the standard (0.0 to 1.0) range. All values below 0.0 will be set to 0.0, and all values above 1.0 will be set to 1.0.

See also

This documentation refers to the Vertex Weight Proximity Modifier specific to the Grease Pencil object. For uses with other object types refer to the general Vertex Weight Proximity Modifier.

Options

Vertex Group
: The vertex group to affect.

  Invert <-->
  : Inverts the influence of the selected vertex group. The setting reverses the weight values of the group.

Target Object
: The object from which to compute distances.

Lowest
: Distance mapping to 0.0 weight.

Highest

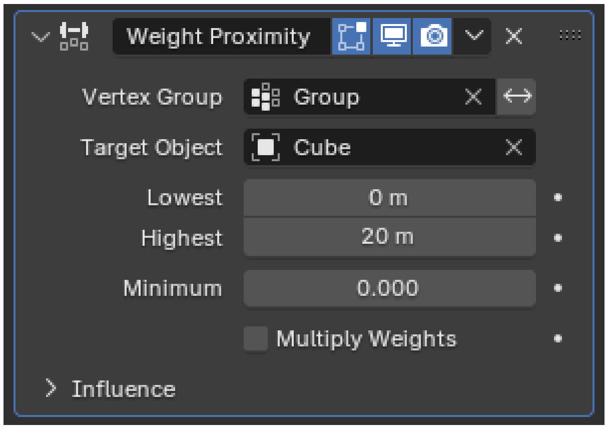
Distance mapping to 1.0 weight.

Minimum

Minimum value for vertex weight.

Multiply Weights

Multiply the calculated weights with the existing values in the vertex group.



Vertex Group: Group

Target Object: Cube

Lowest: 0 m

Highest: 20 m

Minimum: 0.000

☐ Multiply Weights

The Vertex Weight Proximity modifier panel.

Influence

See Influence Filters.

Visual Effects

- Introduction
  - Interface

Types

- Blur Visual Effect
- Colorize Visual Effect
- Flip Visual Effect
- Glow Visual Effect
- Pixelate Visual Effect
- Rim Visual Effect
- Shadow Visual Effect
- Swirl Visual Effect
- Wave Distortion Visual Effect

Introduction

Reference

Panel:: *Properties ‣ Visual Effects*

Grease Pencil has a special set of viewport real-time visual effects that can be apply to the object.

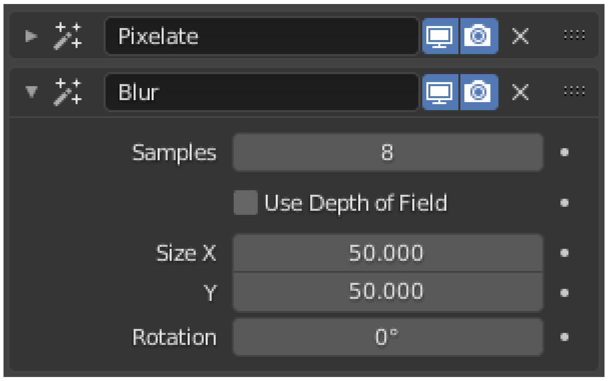
These effects treat the object as if it was just an image, for that reason they have effect on the whole object and cannot limit their influence on certain parts like layers, materials or vertex group as with modifiers. Also unlike modifiers, they can not be applied to the object.

Their main purpose is to have a quick way to apply visual effects on your drawings like blurring, pixelation, wave distortion, among others.

Note

Visual Effects best fit for quick viewport visualization. You can use it for final renders but if you want more precision with effects it is still recommended to use the Compositor.

Interface



Pixelate

Blur

Samples 8

☐ Use Depth of Field

Size X 50.000

Y 50.000

Rotation 0°

Panel layout (Blur effect as an example).

The visual effects panels and interface are similar to modifiers. Each effect shares the same basic interface components similar to modifiers for meshes.

See Modifiers Interface for more information.

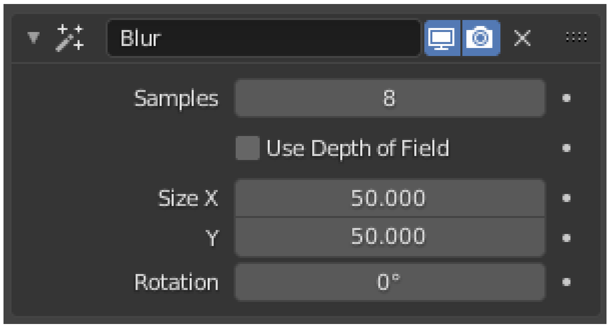
Blur Visual Effect

The *Blur* Visual Effect applies a Gaussian blur to the object.

Options

Samples
  Number of blur samples (0 disabled the blur effect).

Use Depth of Field
  When enabled, the blur effect uses the focal plane distance of the actual camera to calculate the object blur. Only available in camera view.



Blur

Samples 8

☐ Use Depth of Field

Size X 50.000

Y 50.000

Rotation 0°

Blur Visual Effect.

Size X, Y
  Control the blur scale in pixels on the X and Y axis.

Rotation
  Control the Rotation of the blur.

Example

Blur Effect samples (Samples: 8).

Original Model.

Factor: 10, 10.

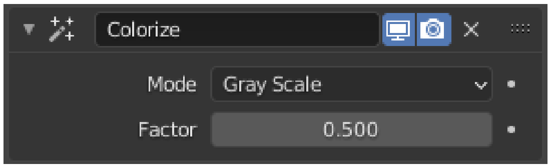
Factor: 50, 50.

Factor: 100, 100.

Colorize Visual Effect

The *Colorize* Visual Effect applies different preset colorizing effects to the object.

Options



Colorize

Mode Gray Scale

Factor 0.500

Blur Visual Effect.

Mode
  Grayscale
    Converts to a grayscale image.

  Sepia
    Converts to a sepia tone image.

  Duotone
    Converts to a posterize image with high contrast and brightness.

    Low Color
      Primary color.

    High Color
      Secondary color.

  Transparent
    Add color transparency.

  Custom
    Allows to define a tint custom color.

    Color
      Sets the tint color.

Factor
  Control the mix value.

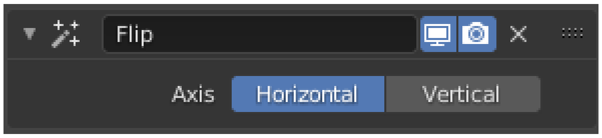
Example

Colorize Effect samples.

Flip Visual Effect

The *Flip* Visual Effect shows the object flipped horizontally and/or vertically.

Options



Flip Visual Effect.

Axis

Which axis or axes to flip the object about.

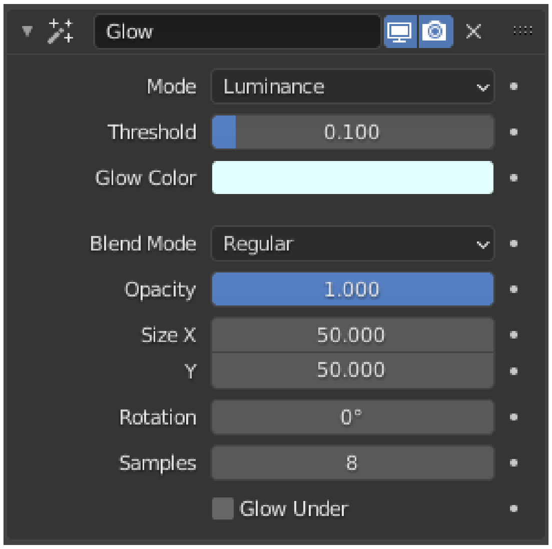
Horizontal:: When enabled, shows the object flipped horizontally.

Vertical:: When enabled, shows the object flipped vertically.

Glow Visual Effect

The *Glow* Visual Effect add a glowing rim around the object.

Options



Glow

Mode Luminance

Threshold 0.100

Glow Color

Blend Mode Regular

Opacity 1.000

Size X 50.000

Y 50.000

Rotation 0°

Samples 8

☐ Glow Under

Glow Visual Effect.

Mode

Determines the mode for the glow effect.

Luminance::	The glow light illuminates the entire object.
Color::	The glow light only affect a single color. Select Color Allows to select a single color to apply the glow light.

Threshold

Limits the colors affected by the glow light. (A value of 1 means no colors affected.)

Glow Color

Defines the glow color.

Blend Mode

The mask blending operation to perform. See Color Blend Modes.

Opacity

Control the Opacity of the glow over the object.

Size X, Y

Control the glow scale in pixels on the X and Y axis.

Rotation

Control the Rotation of the glow.

Samples

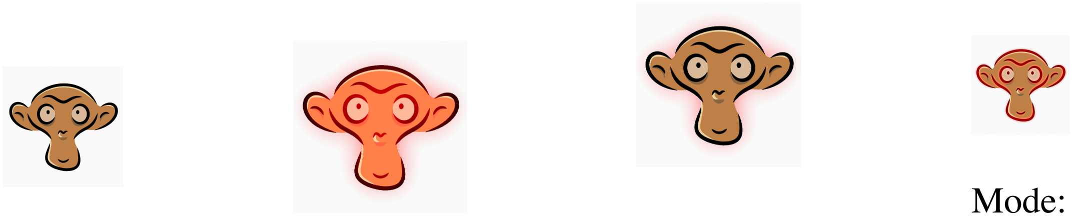
Number of Blur samples (0 disabled the blur effect).

Glow Under

When enabled, glow only affects alpha areas.

Example

Glow Effect samples.



Original image.

Mode: Luminance.

Mode: Luminance (Glow Under).

Mode: Color (Black lines).

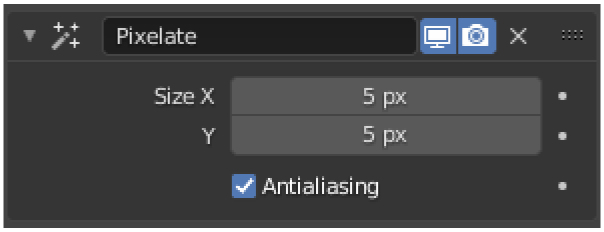
Pixelate Visual Effect

The *Pixelate* Visual Effect shows the object as a pixelated image.

Options

Size X, Y
: Horizontal and vertical size of the final pixels to apply.

Anti-aliasing
: Applies an anti-aliasing effect to the resulting pixels.


Pixelate

Size X 5 px

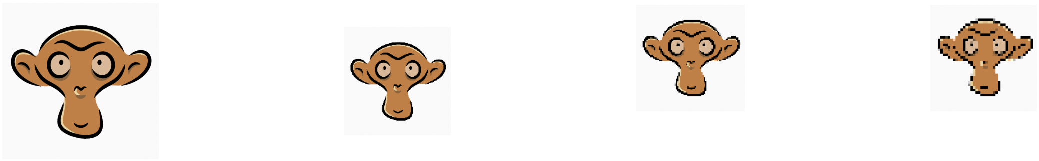
Y 5 px

☑ Antialiasing

Pixelate Visual Effect.

Example

Pixelate Effect samples.



Original image.

Size: 20 px.

Size: 100 px.

Size: 200 px.

Rim Visual Effect

The *Rim* Visual Effect shows a simulated rim light on the object contour.

For simulating the rim light, a masked color silhouette of the object is displaced in horizontal and/or vertical direction.

Many blending modes can be applied to the resulting mask.

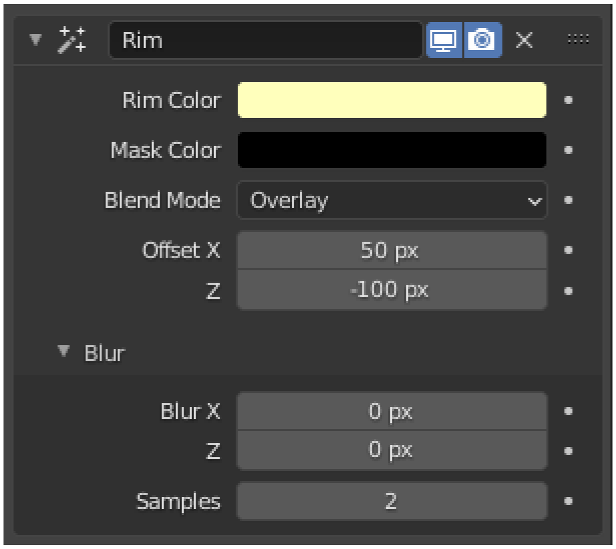
Options

Rim Color
: Defines the rim light color.

Mask Color
: Defines a color to keep unaltered.

Blend Mode
: The mask blending operation to perform. See Color Blend Modes.

Offset X, Y
: Control the color mask displacement in pixels on the X and Y axis.



Rim

Rim Color

Mask Color

Blend Mode: Overlay

Offset X: 50 px

Z: -100 px

Blur

Blur X: 0 px

Z: 0 px

Samples: 2

Rim Visual Effect.

Blur

Blur X, Y
: Control the blur scale in pixels on the X and Y axis.

Samples
: Number of blur samples (0 disabled the blur effect).

Example

Rim Effect samples (Mode: Add).

Original image.

No blur.

Blur.

Mask color: Black.

Shadow Visual Effect

The *Shadow* Visual Effect shows a simulated shadow casting by the object.

For simulating the shadow a color silhouette of the object is displaced in horizontal and/or vertical direction on the back of the object.

Options

Shadow Color

Defines the shadow color.

Offset X, Y

Control the shadow displacement in pixels on the X and Y axis.

Scale X, Y

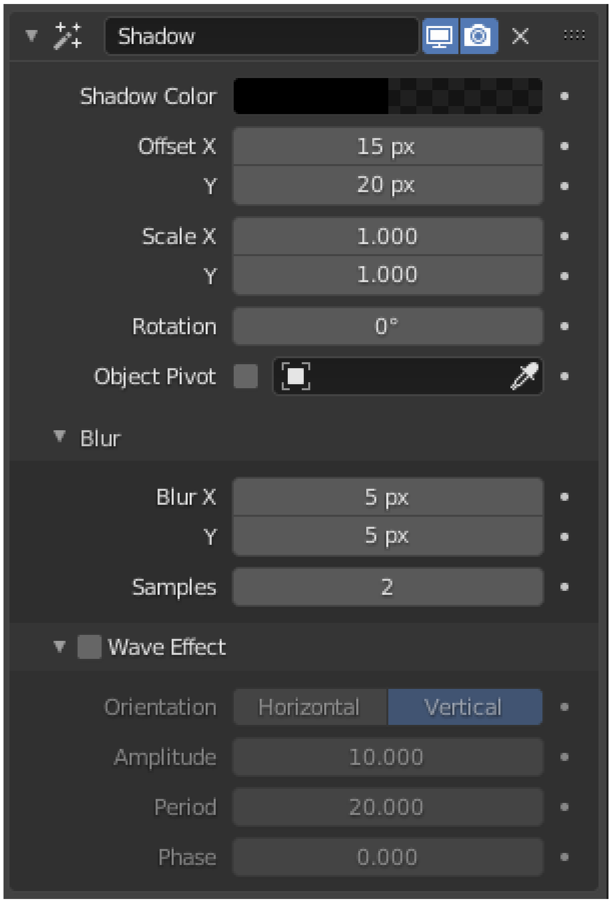
Control the size of the shadow on the X and Y axis.

Rotation

Sets the shadow rotation around the Grease Pencil object center or another object when *Use Object As Pivot* is enabled.

Object Pivot

When enabled, an *Object* is used by the shadow as the center of rotation.



Shadow

Shadow Color

Offset X: 15 px
Y: 20 px

Scale X: 1.000
Y: 1.000

Rotation: 0°

Object Pivot: ☐

Blur

Blur X: 5 px
Y: 5 px

Samples: 2

☐ Wave Effect

Orientation: Horizontal / Vertical (Vertical selected)

Amplitude: 10.000

Period: 20.000

Phase: 0.000

Shadow Visual Effect.

Blur

Blur X, Z
Control the blur scale in pixels on the X and Z axis.

Samples
Number of blur samples (0 disabled the blur effect).

Wave Effect

When enabled, apply a wave distortion to the shadow.

Orientation
Sets horizontal or vertical direction for the waves.

Amplitude
Controls the strength and the depth of the wave.

Period
Controls the wave period. The time it takes to complete one cycle.

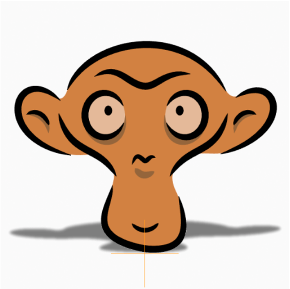
Phase
Shifts the wave pattern over the shadow.

Example

Shadow Effect samples.

Simple Shadow.

Blurred Shadow.



Stretched shadow with an empty as center of rotation.

Swirl Visual Effect

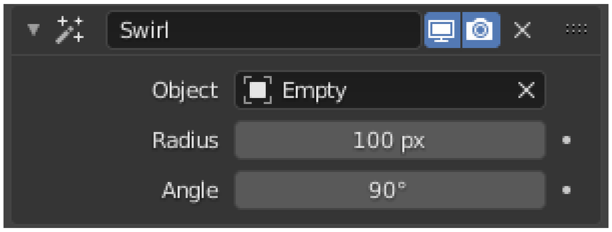
The *Swirl* Visual Effect applies a swirling pattern to the object. The effect use an object as the center of the swirl.

Options

Object
    Sets the object to use as the center of the swirl.

Radius
    External radius size of the swirl.

Angle
    Rotation angle of the swirl. A value of 0 shows no swirl.

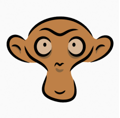


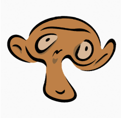
Object: Empty
Radius: 100 px
Angle: 90°

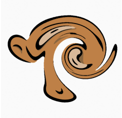
Swirl Visual Effect.

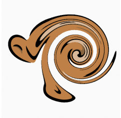
Example

Swirl Effect samples (with a Radius of 100 px).


Angle: 0°.


Angle: 15°.

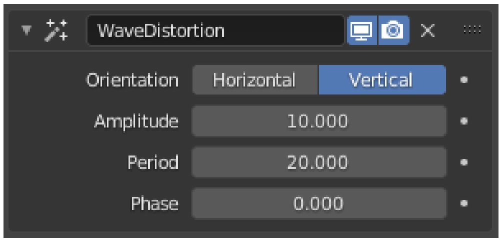

Angle: 45°.


Angle: 90°.

Wave Distortion Visual Effect

The *Wave Distortion* Visual applies a wavy effects to the object.

Options



WaveDistortion

Orientation: Horizontal | Vertical

Amplitude: 10.000

Period: 20.000

Phase: 0.000

Wave Distortion Effect.

Orientation

Sets horizontal or vertical direction for the waves.

Amplitude

Controls the strength and the depth of the wave.

Period

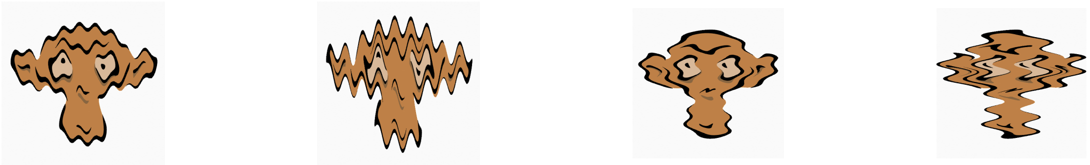
Controls the wave period. The time it takes to complete one cycle.

Phase

Shifts the wave pattern over the Object.

Example

Wave Distortion Effect samples.



Amplitude: 10 (horizontal).

Amplitude: 30 (horizontal).

Amplitude: 10 (vertical).

Amplitude: 30 (vertical).

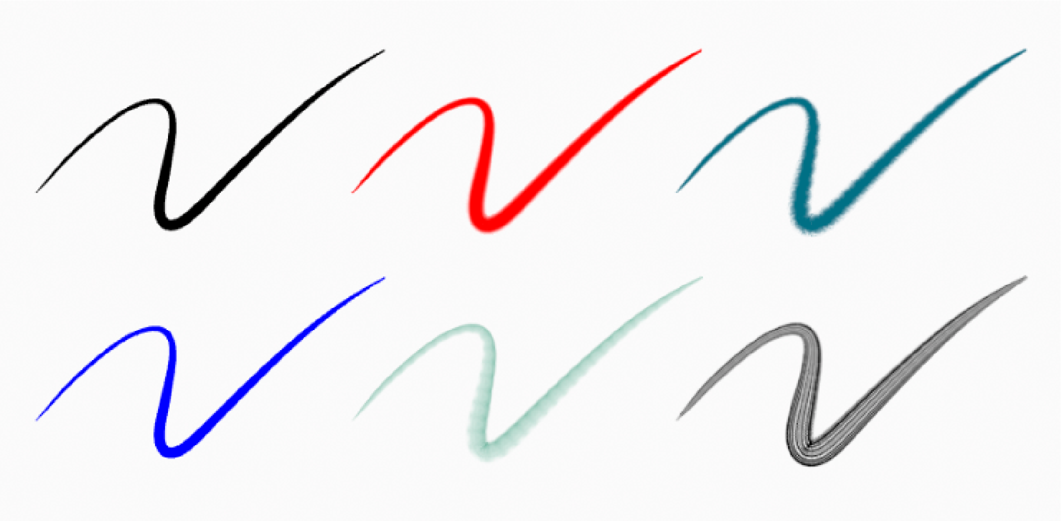
Grease Pencil Materials

Materials control the appearance of the Grease Pencil object. They define the base color and texture of the strokes and filled areas.

There is always only one active material in the list (the selected one). When you draw, the new strokes use the active material.

You can override the base material color using the tools in Vertex Mode or the Draw and Tint tool in Draw Mode.

The material always remains linked to the strokes, this means that any change in a material will change the look of already drawn strokes.



Same stroke linked to different materials.

Material Shader

Grease Pencil materials use a special shader that define the appearance of the surface of the stroke and fill.

Stroke and fill components has it own section panel and they can be enabled with a checkbox on the panel header.

Stroke only has effect on the lines and *Fill* only on the areas determined by closed lines (by connecting the lines start and end points).

Note

The shader is not a BSDF capable shader and can only be setting up on the Material Properties panel (it is not a shader node).

Setting Up Materials

Reference

Mode::	Drawing Mode
Panel::	*Material ‣ Material Slots*
Shortcut::	`U`

Grease Pencil materials can be created in the Material properties as any other materials in Blender. See Material assignment for more information.

The 3D Viewport can be set to Material Preview or Rendered shading, to interactively preview how the material looks in the scene.

Grease Pencil materials are data-blocks that can be assigned to one or more objects, and different materials can be assigned to different strokes.

In Grease Pencil the brush settings together with the material used will define the look and feel of the final strokes.

Materials slots also have some extra controls that help to work with materials while drawing or editing lines.

Properties

Material Slots

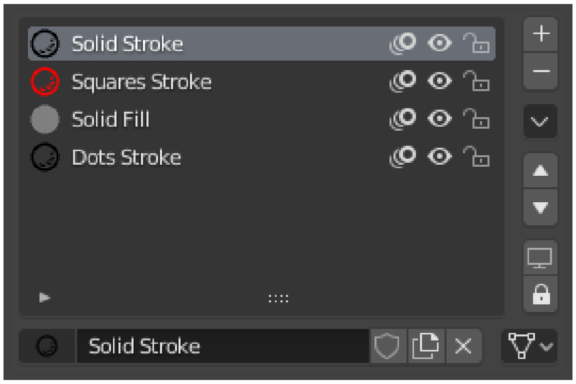
Surface

Settings

Grease Pencil Material Properties

Material Slots

Next to the material name there are three icons buttons that control common properties of the material:



Grease Pencil material slots panel.

/ (Show/Hide in Ghosts)
Toggle the use of the material for [Onion Skinning]().

/ (Hide/Show Material)
Toggle whether the active material is the only one that can be edited and is visible.

/ (Lock/Unlock Material)
Toggle whether the active material is the only one that can be edited.

Specials

Show All
Turns on the visibility of every material in the list.

Hide Others
Turns off the visibility of every material in the list except the active one.

Lock All
Locks edition of all the materials in the list.

Unlock All
Unlocks edition of all the materials in the list.

Lock Unselected
Locks all materials not used in the selected strokes.

Lock Unused

Locks and hides all unused materials.

Copy Material to Selected

Copy the active material to the selected Grease Pencil object.

Copy All Materials to Selected

Copy all materials to the selected Grease Pencil object.

Remove Unused Slots

Remove all unused materials.

Lock & Visibility Controls

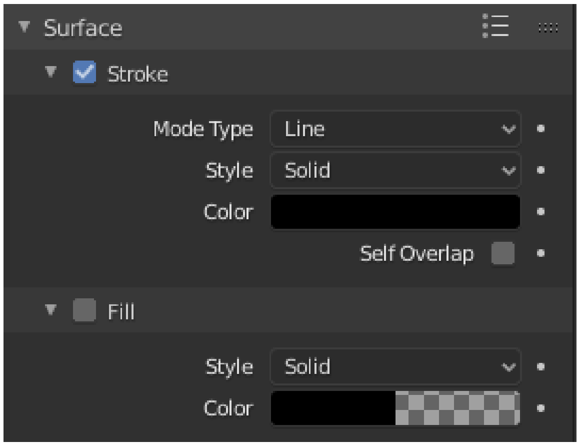
(Isolate Material)

Toggle whether the active material is the only one that can be edited.

(Isolate Material)

Toggle whether the active material is the only one that can be edited and is visible.

Surface


Surface

☑ Stroke

Mode Type: Line

Style: Solid

Color

Self Overlap ☐

☐ Fill

Style: Solid

Color

Shader panel with only Stroke component activated.

Stroke

When enabled, the shader use the stroke component. The *Stroke* component controls how to render the edit lines.

Line Type

Defines how to display or distribute the output material over the stroke.

Line::	Connects every points in the strokes showing a continuous line.
Dots::	Use a disk shape at each point in the stroke. The dots are not connected.
Squares::	Use a square shape at each point in the stroke. The squares are not connected.

Style

The type of the material.

Solid::	Use a solid color.
Texture::	Use an image texture. Image The image data-block used as an image source. Blend Texture and Base Color mixing amount. UV Factor The image size along the stroke.

Base Color

The base color of the stroke.

Holdout

Removes the color from strokes underneath the current by using it as a mask.

Alignment

Defines how to align the *Dots* and *Squares* along the drawing path and with the object's rotation.

Path::	Aligns to the drawing path and the object's rotation.
Object::	Aligns to the object's rotation; ignoring the drawing path.
Fixed::	Aligns to the screen space; ignoring the drawing path and the object's rotation.

Rotation

Rotates the points of *Dot* and *Square* strokes.

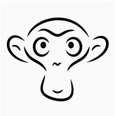
Note

The *Rotation* option is limited to a range of -90 to 90 degrees.

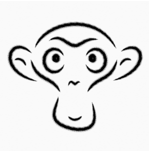
Self Overlap

Disables stencil and overlap self-intersections with alpha materials.

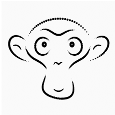
Samples of different material strokes mode types and styles.



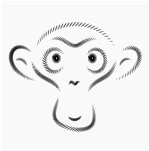
Mode Type: Line, Style: Solid.



Mode Type: Line, Style: Texture.



Mode Type: Dot, Style: Solid.



Mode Type: Dot, Style: Texture.

Fill

When enabled, the shader use the fill component. The *Fill* component control how to render the filled areas determined by closed edit lines.

Style

The type of material.

Solid:: Use solid color.

Gradient:: Use a color gradient.

Gradient Type

- **Linear::** Mix the colors along a single axis.
- **Radial::** Mix the colors radiating from a center point.

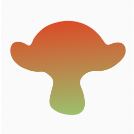
Texture:: Use an image texture.

Image

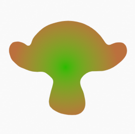
The image data-block used as an image source.

Samples of different material fill styles.

Style: Solid.



Style: Gradient (Linear).



Style: Gradient (Radial).

Style: Texture.

Base Color

The base color of the fill.

Secondary Color *Gradient*

The secondary color.

Holdout

Removes the color from strokes underneath the current by using it as a mask.

Blend *Gradient / Texture*

The amount that the *Secondary Color* (for *Gradient Style*) or image texture (for Texture Style) Mixxes with the *Base Color*.

Flip Colors *Gradient*

Flips the gradient, inverting the Base Color and *Secondary Color*.

Location X, Y *Gradient / Texture*

Shifts the position of gradient or image texture.

Rotation *Gradient / Texture*

Rotates the gradient or image texture.

Scale X, Y *Gradient / Texture*

Scales the gradient or image texture.

Clip Image *Texture*

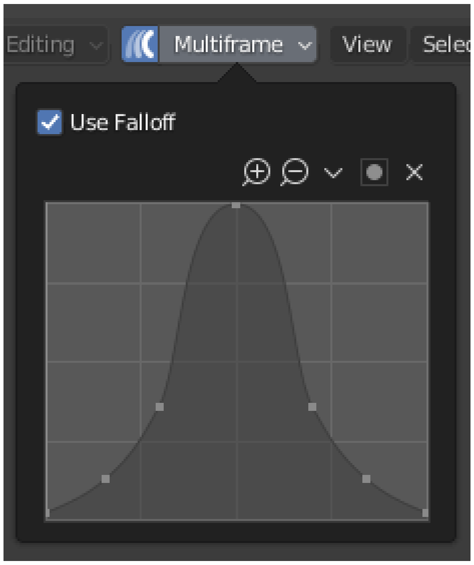
When enabled, show one image instance only (do not repeat).

Settings

Pass Index
: This index can be used with some modifiers to restrict changes to only a certain material. See Modifiers for more information.

Multiframe

Multiframe allows you to draw, edit, sculpt, or weight painting on several frames at the same time. Extremely useful to avoid repeating a task one frame at a time when animating.



Multiframe pop-over.

Use Falloff

When enabled, the effects on the strokes start to falloff from the current frame as defined by a curve widget.

Usage

1. Select the desired keyframes to draw, edit or sculpt at the same time.
2. Activate the Multiframe tool in the 3D Viewport's header with the toggle button (faded lines icon).
3. Once activated you can:
    - Select the points in all the selected keyframes and make your editions.
    - Start sculpting. The sculpt brushes will affects all the strokes in the selected keyframes.
    - Start weight painting. The weight paint brush will affect all the strokes in the selected keyframes.
    - Start Drawing. The new strokes will be added in all the selected keyframes. If you are using the Fill tool then it will be applied in all the selected keyframes.

    - When interpolating you can select the stroke from the different frames in the right order. Interpolate tool will use the selection order to calculate the correct stroke pairs.

Note

Not all operators support Multiframe mode.

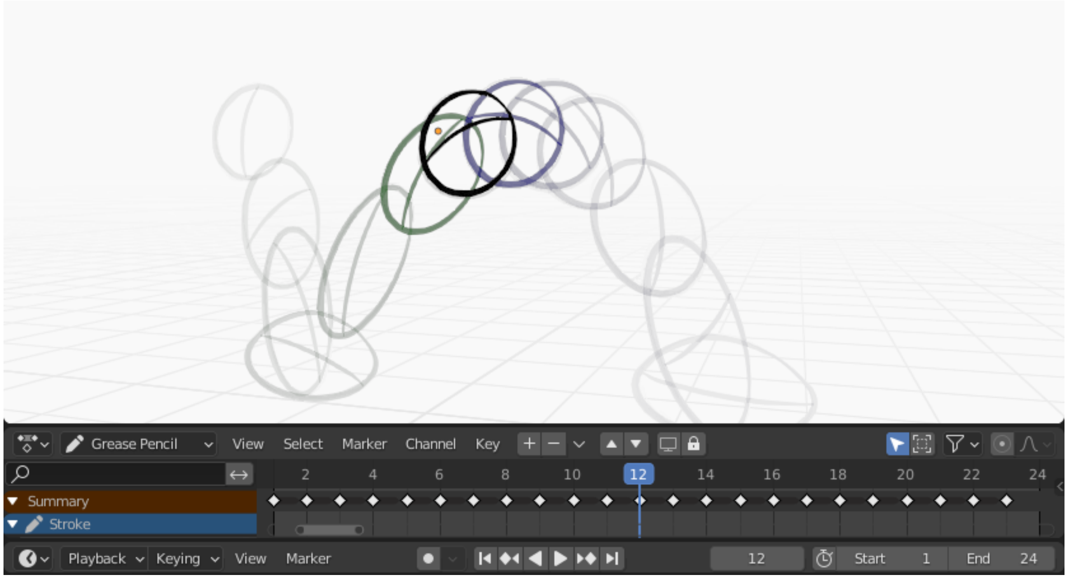
Animation

- Introduction
  - Animating with Grease Pencil
  - 2D Traditional Animation
  - Animation Options
  - Examples
- Interpolation
  - Interpolate
  - Interpolate Sequence
- Animation Tools
  - Insert Blank Keyframe (Active Layer)
  - Insert Blank Keyframe (All Layers)
  - Duplicate Active Keyframe (Active Layer)
  - Duplicate Active Keyframe (All Layers)
  - Delete Active Keyframe (Active Layer)
  - Delete Active Keyframe (All Layers)
  - Interpolate Sequence
  - Bake Mesh to Grease Pencil
  - Bake Object Transform to Grease Pencil

Introduction

Animating with Grease Pencil

The main goal of Grease Pencil is to offer a 2D animation tool full immersed in a 3D environment.


Grease Pencil
View
Select
Marker
Channel
Key
2
4
6
8
10
12
14
16
18
20
22
24
Summary
Stroke
Playback
Keying
View
Marker
12
Start
1
End
24

Sample animation showing Grease Pencil object keyframes in the Dope Sheet with onion skinning enabled.

In Blender, Grease Pencil objects can be animated in many ways:

Moving as a whole object
: Changing their position, orientation or size in time;

Drawing frame by frame
: Drawing one frame at a time (traditional animation).

Deforming them
: Animating their points;

Inherited animation

Causing the object to move based on the movement of another object (e.g. its parent, hook, armature, etc.). Useful for cut-out animation for example.

For a complete overview of animation in Blender please refer to the Animation & Rigging chapter.

2D Traditional Animation

Keyframes

Traditional animation in Grease Pencil is achieved with the use of keyframes that hold the strokes information at a particular frame or frame range.

With Auto keyframe activated, every time you create a stroke in Grease Pencil object Draw Mode a new keyframe is added at the current frame on the active channel. With Auto keyframe deactivated, you will have to add manually a new keyframe or your new strokes will be added on the active keyframe.

See Keyframe Editing for more information.

Note

The channels in the Dope Sheet correspond to the active 2D layer of the Grease Pencil object.

Grease Pencil has its own mode in the Dope Sheet to work with keyframes. See Grease Pencil mode in the Dope Sheet section for more information. There are also several tools on the Stroke menu to work with keyframes and strokes. See Animation tools for more information.

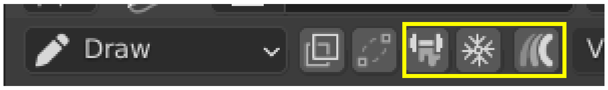
Onion Skinning

One key element in traditional animation is the use of onion skinning. Grease Pencil offer a lot of flexibility and options for this tool. See Onion Skinning for more information.

Animation Options

Draw Mode

In Draw Mode there are three options related to the animation workflow that you can use.



General drawing/animation options.

Add Weight Data

When enabled, new strokes weight data is added according to the current vertex group and weights. If there is no vertex group selected, no weight data is added.

This is useful for example in cut-out animation for adding new drawing on the same vertex group without the need to creating it afterwards.

See Weight Paint Mode for more information.

Additive Drawing

When creating new frames, the strokes from the previous/active frame are include as a basis for the new one.

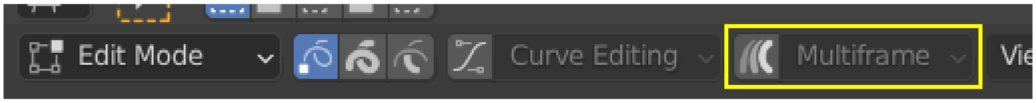
Multiframe

If you need to add new strokes to your animation on several frames you can use multiframe drawing.

You can activate multiframe drawing with the Multiframe button next to the modes selector (faded lines icon). See Multiframe for more information.

Edit Mode

In Edit Mode there is an option related to the animation workflow that you can use.



Multiframe edition.

Multiframe
: Sometimes you may need to modify several frames at the same time with edit tools, for example to repositioning drawings in an animation.

  You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See Multiframe for more information.

Examples

Traditional Animation

This example shows you how to animate a bouncing ball with a traditional 2D animation technique and Grease Pencil.

First, go to menu *File* ‣ *New* ‣ *2D Animation* to start with a new 2D animation template. The template is ready to quick start your animation with a Grease Pencil object already created, Onion Skinning activated, Auto Keyframe enabled and in camera view.

1. Set the range of the animation in the Timeline from 1 to 24.
2. In the 3D Viewport draw a ball on the upper left corner with the Draw Tool (extreme).
3. Move to frame 12 and draw a squashed ball in the bottom center (breakdown).
4. Move to frame 24 and draw a ball in the top right corner of the 3D Viewport (extreme).

5. Keep drawing all the in-between frames you want using the onion skinning ghost as a reference.

To test the animation, press `Spacebar` to play.

Interpolation

Interpolate

Reference

Mode::	Draw and Edit Modes
Tool::	*Toolbar ‣ Interpolate*
Shortcut::	`Ctrl-E`

When you are animating simple shapes you can use the interpolate tool to automatically add new breakdown keyframes.

See Interpolate tool for more details.

Interpolate Sequence

Reference

Mode::	Draw and Edit Modes
Menu::	*Header ‣ Interpolate*
Shortcut::	`Shift-Ctrl-E`

Interpolate strokes between the previous and next keyframe by adding *multiple* keyframes. When you are on a frame between two keyframes and click the sequence button a breakdown keyframe will be added on every frame between the previous and next keyframe.

Step
: The number of frames between generated interpolated frames.

Layer
: Restrict the interpolation to Active or All layers.

Only Selected *Edit Mode*
When enabled, only selected strokes will be interpolated.

Exclude Breakdowns
Exclude existing Breakdowns keyframes as interpolation extremes.

Flip Mode
Invert strokes start and end. Automatic will try to found the right mode for every stroke.

Smooth
Amount of smoothing to apply to interpolated strokes for reducing jitter/noise.

Iterations
Number of time to smooth newly created strokes.

Type
Interpolation method to use for the sequence.

Animation Tools

Insert Blank Keyframe (Active Layer)

Reference

Mode::	Draw Mode, Edit Mode, Sculpt Mode
Menu::	*Stroke ‣ Animation ‣ Insert Blank Keyframe (Active Layer)*
Shortcut::	`Shift-I`

Add a new blank keyframe to the active layer at the current frame. If there is already a keyframe at the current frame, a new blank keyframe will be added on the next frame.

All Layers
    When enabled, Blank keyframe will be created on all layers, not only the active one.

Duration
    The number of blank frames to insert.

Insert Blank Keyframe (All Layers)

Reference

Mode::	Draw Mode, Edit Mode, Sculpt Mode
Menu::	*Stroke ‣ Animation ‣ Insert Blank Keyframe (All Layers)*

Same as Insert Blank Keyframe (Active Layer) but *All Layers* is enabled by default.

Duplicate Active Keyframe (Active Layer)

Reference

Mode::	Draw Mode, Edit Mode, Sculpt Mode
Menu::	*Stroke ‣ Animation ‣ Duplicate Active Keyframe (Active Layer)*

Duplicates the strokes on the last keyframe by copying them to the current frame.

Mode
Pick which layers to duplicate.

Active::	Duplicate only the active layer.
All::	Duplicate all the layers.

Duplicate Active Keyframe (All Layers)

Reference

Mode::	Draw Mode, Edit Mode, Sculpt Mode
Menu::	*Stroke ‣ Animation ‣ Duplicate Active Keyframe (All Layers)*

Same as Duplicate Active Keyframe (Active Layer) but the *Mode* is set to *All* by default.

Delete Active Keyframe (Active Layer)

Reference

Mode::	Draw Mode, Edit Mode, Sculpt Mode
Menu::	*Stroke ‣ Animation ‣ Delete Active Keyframe (Active Layer)*

Shortcut:: `Alt-I`

Deletes the last keyframe in the Dope Sheet or the current keyframe if you are on one.

Type
    Pick which layer to delete keyframes.

Active Frame:: Deletes current frame in the active layer.
All Active Frames:: Delete active frames for all layers.

Delete Active Keyframe (All Layers)

Reference

Mode:: Draw Mode, Edit Mode, Sculpt Mode
Menu:: *Stroke ▸ Animation ▸ Delete Active Keyframes (All Layers)*
Shortcut:: `Shift-Delete`

Same as Duplicate Active Keyframe (Active Layer) but the *Type* is set to *All Active Frames* by default.

Interpolate Sequence

Reference

Mode:: Draw Mode, Edit Mode
Menu:: *Grease Pencil ▸ Interpolate Sequence*
Shortcut:: `Shift-Ctrl-E`

Interpolate strokes between the previous and next keyframe by adding *multiple* keyframes. A breakdown keyframe will be added on every frame between the previous and next keyframe.

Step
umber of frames between generated interpolated frames.

Layer
Layers included in the interpolation.

Exclude Break Downs
Exclude existing Breakdowns keyframes as interpolation extremes.

Flip Mode
Invert destination stroke to match start and end with source stroke.

Smooth
Amount of smoothing to apply to interpolated strokes, to reduce jitter/noise.

Iterations
Number of times to smooth newly created strokes.

Type
Interpolation method to use the next time *Interpolate Sequence* is run.

Bake Mesh to Grease Pencil

Reference

Editor::	3D Viewport
Mode::	Object Mode
Menu::	*Object* ‣ *Animation* ‣ *Bake Mesh to Grease Pencil…*

Converts each frame of a mesh animation within a selected frame range to a Grease Pencil object keyframed strokes. The *Bake Action* operator computes the final animation of the selected objects with all those modifiers, drivers, and constraints applied, and keyframes the result.

Target Object

Determines if the image empty is kept or replaced.

New Object: Creates a new grease pencil object and keeps the image empty Selected Object: Replaces the image empty with the grease pencil object.

Start Frame, End Frame

Start/End frame for the baking process.

Step

Frame steps for the baking process

Thickness

Strokes thickness.

Threshold Angle

Threshold value that determine the strokes end.

Stroke Offset

Sets offset to separate strokes from filled strokes.

Only Seam Edges

Convert only edges marked as seam.

Export Faces

Convert faces as filled strokes.

Only Selected Keyframes

Convert only the selected keyframes.

Target Frame

Target destination frame for the baked animation.

Projection Type

Sets the projection type to use for the converted strokes.

Bake Object Transform to Grease Pencil

Reference

Editor::	3D Viewport
Mode::	Object Mode
Menu::	*Object ‣ Animation ‣ Bake Object Transform to Grease Pencil*

Applies all transform animation at Object level within a selected frame range to Grease Pencil object keyframes.

Start Frame, End Frame
    Start/End frame for the baking process.

Step
    Frame steps for the baking process.

Only Selected Keyframes
    Convert only the selected keyframes.

Target Frame
    Target destination frame for the baked animation.

Projection Type
    Sets the projection type to use for the converted strokes.

Grease Pencil Object Modes

Object Modes allow editing different aspects of Grease Pencil objects. These modes are specifically tailored to the Grease Pencil object, unlike the more general modes, which work for other object types (with the exception of object mode which is the same for all objects).

- Draw Mode

Draw Mode is where new strokes are created. Strokes are directly sketched on a canvas, using different tools and brushes.

- Sculpt Mode

Sculpt Mode can be used to deform and shape existing strokes more organically. Strokes can be smoothed, deformed, or reshaped, adding fluidity and dynamism to drawings.

- Edit Mode

Edit Mode allows modifying individual strokes and points of Grease Pencil objects. This mode is ideal for fine-tuning linework, adjusting shapes, and refining details.

- Vertex Paint Mode

Vertex Paint Mode allows adding color the vertices of strokes directly. This mode is useful for adding shading, gradients, or detailed color effects providing finer control over the drawing’s appearance.

- Weight Paint Mode

Weight Paint Mode allows assigning vertex weights to strokes. This is crucial for rigging and animating characters, ensuring smooth, and precise deformations based on the painted weights.

- Object Mode

This mode allows working with the entire Grease Pencil object as a whole. It’s used for overall transformations and managing the placement of the object within the scene.

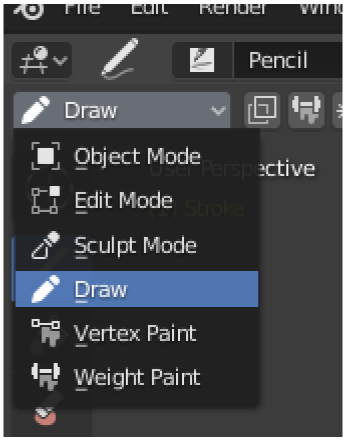
Draw Mode

- Introduction
    - Strokes Location & Orientation Controls
    - Drawing Options
- Brushes
    - Draw Brushes
    - Fill Brushes
    - Erase Brushes
- Drawing Tools
- Tools
    - Brush Tool
    - Erase Tool
    - Fill Tool
    - Trim Tool
    - Eyedropper
    - Line Tool
    - Polyline Tool
    - Arc Tool
    - Curve Tool
    - Box Tool
    - Circle Tool
    - Interpolate
- Tools Settings
    - Brush Asset
    - Brush Settings
    - Color
- Stroke Placement
- Drawing Plane
- Drawing Operations
    - Active Layer
    - Animation
    - Interpolate Sequence

Introduction

Draw Mode is the mode in Grease Pencil that allows you to draw in the 3D Viewport. This mode is actually the only one in which new strokes can be created.

Already made strokes can not be selected in Draw Mode, for editing strokes you must use the Edit Mode or Sculpt Mode.



3D Viewport Mode selector: Draw Mode.

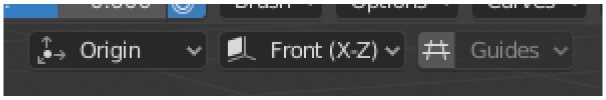
Draw Mode is selected with the *Mode* menu in the 3D Viewport header. Once Draw Mode is activated, the Toolbar of the 3D Viewport will change to Draw Mode specific panels. Also a circle with the same color as the active material will appear and follow the location of the cursor in the 3D Viewport.

To create new strokes you have to select one of the drawing tools in the Toolbar. The most common one is the Draw tool for free-hand drawings but there are many other tools for drawing, filling areas and erasing strokes. There are also some tools to create primitives shapes like lines, arcs, curves, boxes and circles.

See Toolbar for more details.

Strokes Location & Orientation Controls

Drawing in a 3D space is not the same as drawing on a flat canvas. When drawing with Grease Pencil you have to define the location and orientation of the new strokes in the 3D space.



3D Viewport header Controls for strokes.

Stroke Placement

The Stroke Placement selector defines the new strokes location in 3D space.

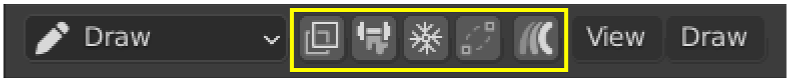
See Stroke Placement for more information.

Drawing Planes

The Drawing Planes selector defines the plane (orientation) to which the new strokes will be restricted.

See Drawing Planes for more information.

Drawing Options



General drawing options.

Multiframe

Allows to draw on several frames at the same time.

See Multiframe for more information.

Additive Drawing

When creating new frames adding strokes with drawing tools, the strokes from the previous/active frame are include as a basis for the new one. When erasing existing strokes using Additive Drawing a new keyframe will be added.

AutoMerge

Joins new strokes with the beginning or end of previously drawn strokes in the active layer.

Add Weight Data

When enabled, weight data is added to new strokes according to the current vertex group and weight. If there is no vertex group selected, no weight data is added.

Useful for example in cut-out animation for adding new drawing on the same vertex group without the need to creating it afterwards.

See Weight Paint Mode for more information.

Draw on Back

When enabled, new strokes are drawn below of all strokes in the layer. For example when you want to paint with a fill material below line strokes on a character and they are on the same layer.

Brushes

There are a number of brushes for draw mode bundled in the *Essentials* asset library. This is an overview of all of them.

Draw Brushes

Draw brushes are the special type of brushes that uses Grease Pencil for drawing tools. The brush can be changed in the Tool Settings. The different draw brushes (pencil, Ink, marker, etc.) are settings variations of the same *Draw Brush*. You can create many brushes, each with unique settings to get different artistic result while drawing.

Fill Brushes

Fill brushes are the special type of brushes that uses Grease Pencil for the *Fill* tools. The brush can be changed in the Tool Settings. The different fill brushes are settings variations of the same *Fill Brush*. You can create many brushes, each with unique settings to get different result when filling areas.

Erase Brushes

Erase brushes are the special types of brushes that uses Grease Pencil for *Erase* tools. The brush can be changed in the Tool Settings. Soft and hard eraser brushes are settings variations of the same *Erase Brush*. You can create many brushes, each with unique settings to get different effects while erasing. The *Erase Brush* has also other two special eraser types: point and stroke.

Draw Brushes

Reference

Mode::	Draw Mode
Brush::	*Asset Shelf ‣ Draw*

The Draw brush allows you to draw free-hand strokes.

Brush Settings

Material

Data-block selector for the material.

Radius

The radius of the brush in pixels.

`F` allows you to change the brush size interactively by dragging the pointer or by typing a number then confirm.

(Size Pressure)

Adjusts the radius based on the stylus pressure when using a Graphics Tablet. The gradient of the pressure can be customized using the curve widget.

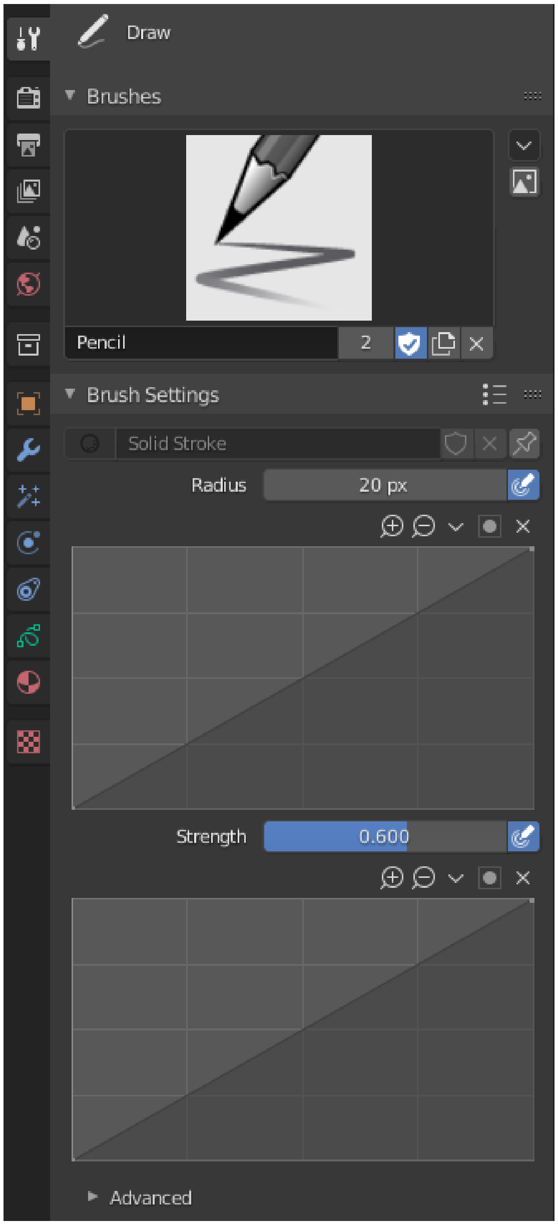
Strength

Control the stroke transparency (alpha). From fully transparent (0.0) to fully opaque (1.0).

You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the pointer and then `LMB`. You can also enter the size numerically.

(Strength Pressure)

Adjusts the strength based on the stylus pressure when using a Graphics Tablet. The gradient of the pressure can be customized using the curve widget.



Radius 20 px

Strength 0.600

Caps Type
: The shape of the start and end of the stroke.

Round:: Strokes start and stop with a curved shape.

Flat:: Strokes start and stop with a straight cutoff.

Advanced

Input Samples
: Controls how often the input device is read to generate points on the stroke. Higher values give a higher precision (more points) but produce an irregular stroke, while lower values give a lower precision (fewer

points) but produce a soften stroke. (0 disabled extra input device samples.)

You have to set up this value according to your input device to obtain the right balance between accuracy and softness for your strokes. See Input Device for more information.

Active Smooth
The number of smoothing iterations to apply to the stroke while drawing.

Angle
Direction of the input device that gives the maximum thickness to the stroke (0° for horizontal).

Factor
Amount of thickness reduction when the stroke is perpendicular to the *Angle* value.

Hardness
Amount of transparency (alpha) to apply from the border of the point to the center. Works only when the brush is using stroke materials of *Dot* or *Box* style.

Aspect X, Y
Controls the width and height of the alpha gradient.

Stroke

Post-Processing

Post-processing methods that are executed on the strokes when you finished drawing, right after releasing the `LMB` or `Pen` tip. You can toggle the use of post-processing using the checkbox in the section panel header.

Smooth
Strength of smoothing process on the points location along the stroke.

Iterations
: The number of smoothing iterations to apply to the stroke.

Subdivision Steps
: Number of subdivisions to apply to newly created strokes.

Simplify
: Reduces final points numbers in the stroke with an adaptive algorithm.

Trim Strokes End
: Automatically trim intersection strokes ends.

Outline
: Activate the conversion of the newly created stroke to its outline.

  Material
  : Material used for outline stroke.

  Thickness
  : Thickness used for outline stroke.

Randomize

Adds randomness to the position of the points along the stroke. You can toggle the use of Randomize using the checkbox in the section panel header.

Radius
: The amount of randomness to apply using the pressure of the input device.

Strength
: The amount of randomness to apply to the stroke strength value (alpha).

UV
: The amount of randomness to apply to the UV rotation.

Hue, Saturation, Value
: Randomizes the hue, saturation, and value of the stroke's Color.

Jitter
The amount of jittering to add to the stroke.

Common Options

Stroke Random (stroke icon)
Use randomness only at stroke level.

(Use Pressure)
Uses the stylus pressure to control how strong the effect is. The gradient of the pressure can be customized using the curve widget.

Stabilize Stroke

Stabilize Stroke helps to reduce jitter of the strokes while drawing by delaying and correcting the location of points. You can toggle the use of *Stabilize Stroke* using the checkbox in the section panel header.

Radius
Minimum distance from the last point before the stroke continues.

Factor
A smooth factor, where higher values result in smoother strokes but the drawing sensation feels like as if you were pulling the stroke.

Cursor

The cursor can be disabled by toggling the checkbox in the *Cursor* header.

Show Fill Color while Drawing
Shows the brush linked material color in the viewport.

Tint Brush

Reference

Mode::	Draw Mode
Brush::	*Brush* ▸ *Tint*

The Tint brush allows you to paint onto strokes point mixing the material base color with a selected color.

Brush Settings

Mode

Defines how Color Attributes affect to the strokes.

Stroke and Fill::	Color Attributes affects both the Stroke and Fill materials.
Stroke::	Color Attributes affects the Stroke material only.
Fill::	Color Attributes affects the Fill material only.

Usage

Selecting a Brush, Color & Mode

In the Tool Settings select the brush, color and mode to use with the tool.

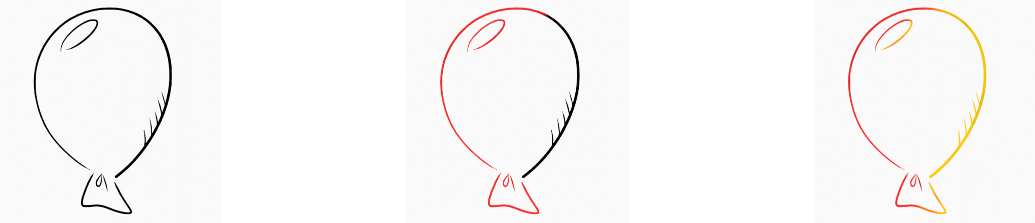
You can configure the brush main settings included in the Tool Settings for convenience. For the vertex paint brushes configuration and settings see Vertex Paint Brush.

`Ctrl-LMB` erase the Color Attribute.

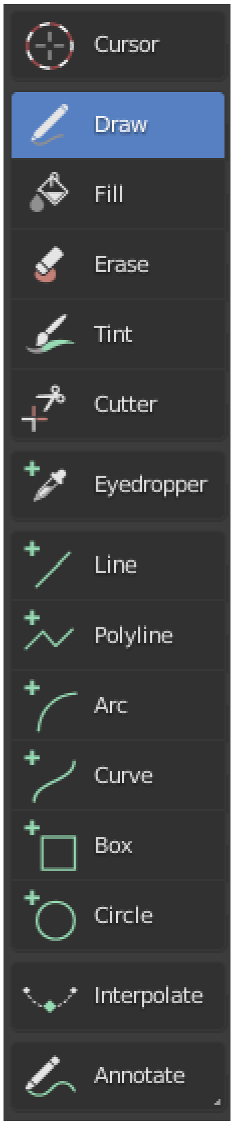
Painting

Click and hold `LMB` or use the pen tip to paint onto the stroke points.

Vertex painting stroke points.



Drawing Tools


Cursor
Draw
Fill
Erase
Tint
Cutter
Eyedropper
Line
Polyline
Arc
Curve
Box
Circle
Interpolate
Annotate

Cursor
Change the location of the 3D Cursor.

Brush
Tool to use for any of the drawing brushes.

Fill
Automatic fill closed strokes areas.

Erase
Erase strokes.

Box
Draw rectangular shapes.

- Circle
  Draw oval shapes.
- Line
  Draw straight lines.
- Polyline
  Draw straight multiple lines.
- Arc
  Draw simple arcs.
- Curve
  Draw complex Bézier style curves.

Trim
Cut strokes in between others.

Eyedropper

Eyedropper to create new materials or palette color based on sampled colors in the 3D Viewport.

Interpolate `Ctrl-E`

Automatically create a breakdown keyframe between two normal keyframes.

Tools

- Brush Tool
  - Tool Settings
  - Usage
- Erase Tool
  - Tool Settings
  - Usage
- Fill Tool
  - Tool Settings
  - Usage
- Trim Tool
  - Tool Settings
  - Usage
- Eyedropper
  - Tool Settings
  - Usage
- Line Tool
  - Tool Settings
  - Usage
- Polyline Tool
  - Tool Settings
  - Usage
- Arc Tool
  - Tool Settings
  - Usage
- Curve Tool
  - Tool Settings
  - Usage
- Box Tool
  - Tool Settings
  - Usage
- Circle Tool
  - Tool Settings
  - Usage

- Interpolate
    - Usage
    - Tool Settings

Brush Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ‣ *Brush*

Tool to free form draw Grease Pencil strokes using any of the *Draw* type brushes.

Tip

Activating a brush asset from an asset shelf or brush selector will also activate this tool for convenience.

Tool Settings

Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Eraser

Default Eraser Brush

Select a brush to use as eraser for quickly alternating with the main brush using `Ctrl-LMB`.

Color

Settings to determine the color of strokes.

See Color

Usage

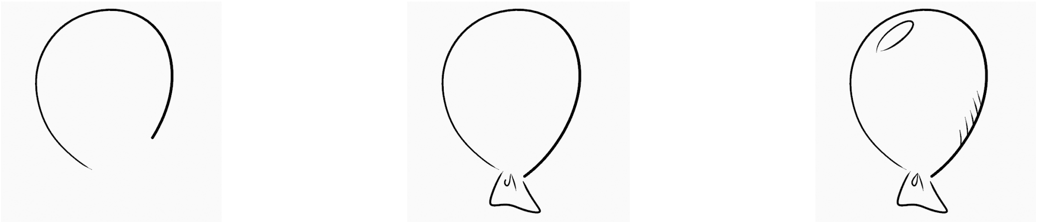
Selecting a Brush and Material

In the Tool Settings select the brush, material and color type to use with the tool. The Draw tool uses *Draw Brush* types. See Brush Settings for more information.

Free-hand Drawing

Click and hold `LMB` or use the pen tip to make free-hand drawing on the viewport.

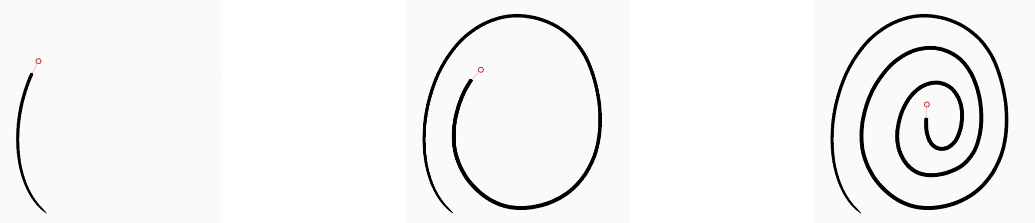
Drawing free-hand strokes.



Stabilize Stroke

`Shift-LMB` toggle the use of Stabilize Stroke on the brush to have more control while drawing and get smoother lines.

Drawing strokes using *Stabilize Stroke*.



Straight Lines

`Alt-LMB` Constrains the drawing of the strokes to horizontal or vertical straight lines.

Switching to the Erase Tool

`Ctrl-LMB` changes temporally to the active Erase tool. See Erase Tool for more information.

You can also use `B` to delete all the points in the selected drawing area.

Erase Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ▸ *Brush*

The Erase tool erases already drawn strokes.

The Erase tool uses any of the Grease Pencil *Erase* draw mode brushes. Activating a brush from an asset shelf or brush selector will also activate this tool for convenience.

Tool Settings

Brush Asset

The asset selector can be used to open a pop-up asset browser to select the active brush asset for the tool.

See Asset Operators for more information.

Brush Settings

Radius

The radius of the brush in pixels.

`F` allows you to change the brush size interactively by dragging the pointer or by typing a number then confirm.

(Size Pressure)

Adjusts the radius based on the stylus pressure when using a Graphics Tablet.

Strength

Control how much will affect the eraser has on the stroke transparency (alpha).

You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the pointer and then `LMB`. You can also enter the size numerically.

(Strength Pressure)

Adjusts the strength based on the stylus pressure when using a Graphics Tablet.

Mode

Determines how the erase tool behaves.

Dissolve::	To simulate a raster type eraser, this eraser type affects the strength and thickness of the strokes before actually delete a point.
Point::	Delete one point at a time.
Stroke::	Delete an entire stroke.

Cursor

The cursor can be disabled by toggling the checkbox in the *Cursor* pop-over menu.

Usage

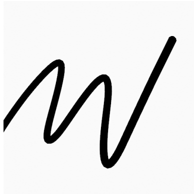
Selecting a Brush

In the Tool Settings select the brush to use with the tool. The Erase tool uses Erase Brush types (soft, point and stroke).

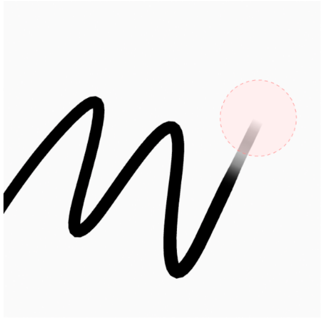
Dissolve Erasing

- Select an erase brush of type Soft/Hard.

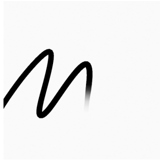
- Adjust brush settings.
- Click and hold `LMB` or use the `Pen` tip to delete strokes on the viewport.



Original drawing.



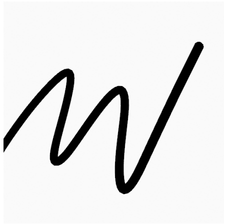
The eraser affect the transparency of the strokes.



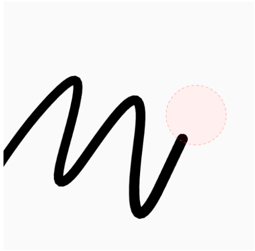
Final result.

Point Erasing

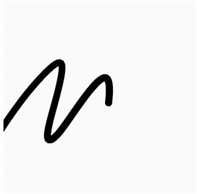
- Select an erase brush of type Point.
- Adjust brush settings.
- Click and hold `LMB` or use the `Pen` tip to delete strokes on the viewport.



Original drawing.



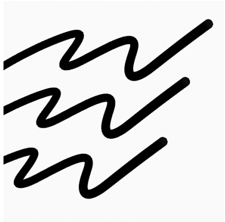
The eraser delete one point at a time.



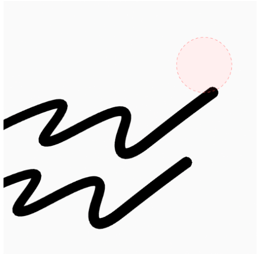
Final result.

Stroke Erasing

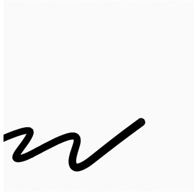
- Select an erase brush of type Stroke.
- Adjust brush settings.
- Click and hold `LMB` or use the `Pen` tip to delete strokes on the viewport.



Original drawing.



The eraser delete one stroke at a time.



Final result.

Fill Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar ‣ Fill*

The Fill tool is used to automatically fill closed strokes areas.

The Fill tool uses any of the Grease Pencil *Fill* draw mode brushes. Activating a brush from an asset shelf or brush selector will also activate this tool for convenience.

Tool Settings

Brush Asset

The asset selector can be used to open a pop-up asset browser to select the active brush asset for the tool.

See Asset Operators for more information.

Brush Settings

You can also configure the brush main settings exposed on the Tool Settings for convenience.

Direction `Ctrl`

The portion of area to fill.

Normal::	Fills the area inside the shape under the cursor.
Inverted::	Fills the area outside the shape under the cursor.

Precision

Multiplier for fill boundary accuracy. Higher values are more accurate but slower.

Dilate/Contract

Size in pixels to expand or shrink the fill area from the strokes boundary.

Thickness

The thickness radius of the boundary stroke in pixels.

Advanced

Boundary

Sets the type of fill boundary limits calculation to perform.

All::	Use the thickness of the strokes and the editing lines together.
Strokes::	Use only the thickness of the strokes (ignore edit lines).
Edit Lines::	Use only the edit lines (ignore strokes).

Show Lines (eye icon)

Toggle show auxiliary lines to see the fill boundary.

Layers

Determines which Layers are used for boundary strokes.

Visible::	Calculates boundaries based on all visible layers.
Active::	Calculates boundaries based on the active layer.
Layer Above::	Calculates boundaries based on the layer above the active layer.
Layer Below::	Calculates boundaries based on the layer below the active layer.
All Above::	Calculates boundaries based on all layers above the active layer.
All Below::	Calculates boundaries based on all layers below the active layer.

Simplify
Number of simplify steps to apply to the boundary line. Higher values reduce the accuracy of the final filled area.

Ignore Transparent
When enabled, strokes with transparency does not take into account on fill boundary calculations.

The value slider controls the threshold to consider a material transparent.

Limit to Viewport
When enabled, fill only visible areas in the viewport.

Gap Closure

Gap closure lines are automatic temporarily lines that help to close gaps on the strokes.

Size
Control the Size of the line extension or the circumference to use to calculate the lines that will close the gaps.

Mode `S`
Sets the type of Gap closure method to use.

Radius::	Uses the Radius of circumference of opened nearest points to calculate a line that close the gap.
Extend::	Extends the opened strokes to close gaps.

Visual Aids
Toggle show closure lines helper.

Strokes Collision `D`
Check if extend lines collide with strokes, stopping the extension if a collision is detected.

Only Collide Lines
: Use for closing gaps only if the extend strokes collide.

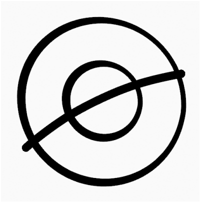
Usage

Selecting a Brush and Material

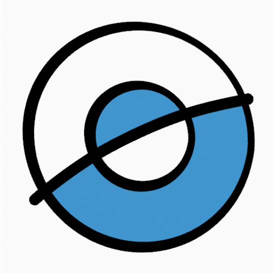
In the Tool Settings select the brush, material and color type to use with the tool. The Fill tool uses *Fill Brush* types. See Brush Settings for more information.

Filling Areas

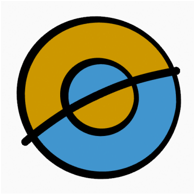
Click `LMB` in a closed stroke area. The tool will automatically calculate the boundary and create a new closed stroke filled with the material selected.



Original Drawing.



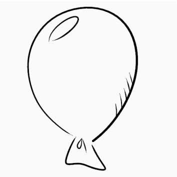
Use the fill tool to leak materials on closed areas.



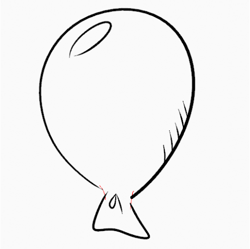
Final filled drawing.

Boundary Strokes

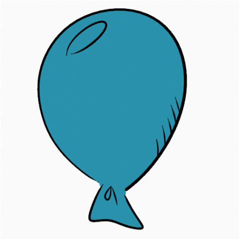
If you have a large gap in an area that you want fill, you can add boundary strokes manually, a temporary auxiliary lines for closing open shapes. To create a boundary stroke use `Alt-LMB` and draw a line to close the desired area.



Original drawing.



Add boundary strokes to close open areas (red lines).



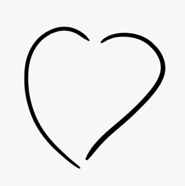
Use the Fill tool to leak material on the new closed area.

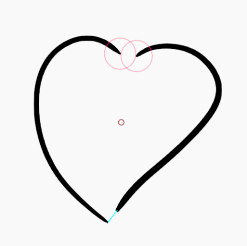
When you are satisfied with the fill result you can delete the boundary strokes using the *Clean Up* tool in the Grease Pencil Menu in Edit Mode.

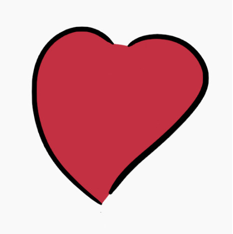
Automatic Gap Closure

A more automatic way to close gaps in an area that you want fill is using temporarily helper lines. There are two method to use “Radius” or “Extend”

Radius use temporary auxiliary lines calculated from the radius of nearby open points to close open shapes. Set the size more than zero to control the circle size over opened points (the circle will disappear when the line close the gap). Click over the area you want to be filled and change the length of the strokes using `PageUp` `PageDown` or `Wheel`. When you are satisfied with the length and you are sure the temporarily strokes cross each other, click again to fill the area.





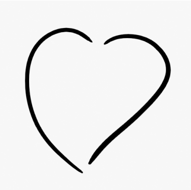


Original Drawing.

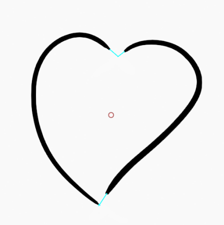
Use Radius mode to close open areas (Red circles and cyan lines).

Use Fill Tool to leak material on the new closed area.

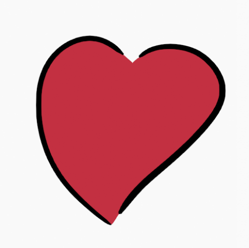
Extend use temporary auxiliary lines extending the actual strokes ends for closing open shapes. Set the size more than zero to use the extended lines, click over the area you want to be filled and change the length of the strokes using `PageUp`/`PageDown`, `Wheel` or a pen's `MMB`. When you are satisfied with the length and you are sure the temporarily strokes cross each other, click again to fill the area.



Original Drawing.



Use Extend mode to close open areas (cyan lines).



Use Fill Tool to leak material on the new closed area.

Trim Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ▸ *Trim*

The Trim tool delete points in between intersecting strokes.

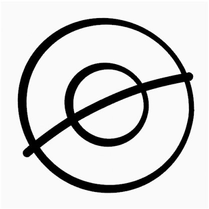
Tool Settings

Flat Caps
    Mark newly created End Caps as *Flat*.

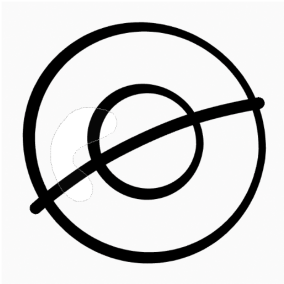
Threshold
    Determine the threshold for stroke intersections.

Usage

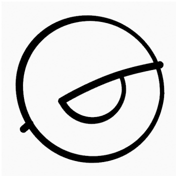
Draw a dotted line around the strokes you want to trim. After releasing the mouse button all the points on the selected strokes will be deleted until another intersecting stroke is found.



Original drawing.



Lasso Selecting the strokes to be



Final result.

trimmed.

Eyedropper

Reference

Mode::	Draw Mode
Tool::	*Toolbar ‣ Eyedropper*

The Eyedropper tool is used to create materials or palette color based on sampled colors in the 3D Viewport.

Tool Settings

Material:: Create a new material with the Stroke Base Color to be the sampled color.

Material Mode

The color transformation will be applied on the stroke and/or the fill color.

Stroke::	Only paint over strokes.
Fill::	Only paint over fill areas.
Both::	Paint over strokes and fill areas

Palette:: Add a new color to the color palette based on the sampled color.

Brush:: Sets the brush color to the sampled color.

Usage

- `LMB` Create a stroke material.
- `Shift-LMB` Create a fill material.
- `Shift-Ctrl-LMB` Create both a stroke and fill material.
- Holding `LMB` and dragging accumulates the average color under the mouse cursor.

Line Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ‣ *Line*

The Line tool create straight lines using any of the *Draw* type brushes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions

The number of stroke points between each stroke edge.

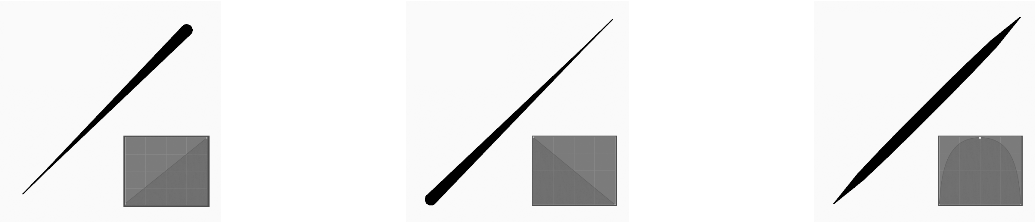
Thickness Profile

Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve

When enabled, the stroke use a curve profile to control the thickness along the line.

Different thickness profile samples.



Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Color

Settings to determine the color of strokes.

See Color

Usage

Selecting a Brush and Material

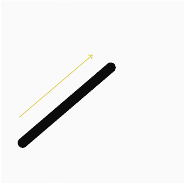
In the Tool Settings select the brush, material and color type to use with the tool. The Line tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Lines

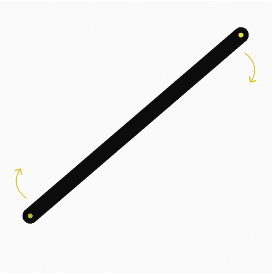
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to snapping the line to horizontal, vertical or 45° angle or use `Alt` to create the line from a center point.

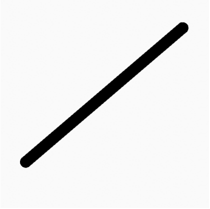
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final line.



click and dragging the start point.



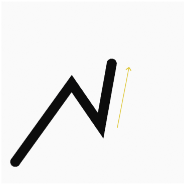
Moving start and end points with manipulators.



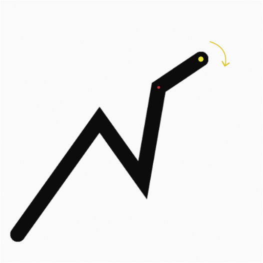
The line after confirming.

Extruding

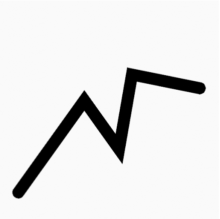
Before confirming you can use `E` to extrude the end point of the line to generate multiple connected lines.



End point extruding.



Moving the end point of the last line with the manipulator.



The connected lines after confirming.

Polyline Tool

Reference

Mode:: Draw Mode
Tool:: *Toolbar ‣ Polyline*

The Polyline tool creates multiple straight lines using any of the *Draw* type brushes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions

The number of stroke points between each stroke edge.

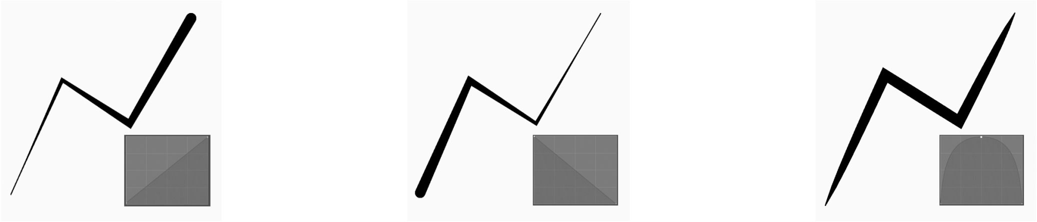
Thickness Profile

Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve

When enabled, the stroke use a curve profile to control the thickness along the line.

Different thickness profile samples.



Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Color

Settings to determine the color of strokes.

See Color

Usage

Selecting a Brush and Material

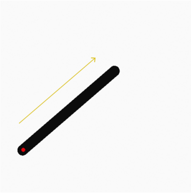
In the Tool Settings select the brush, material and color type to use with the tool. The Line tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Polylines

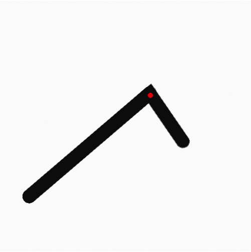
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. Click multiple times on different locations to create multiple connected lines.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to snapping the line to horizontal, vertical or 45° angle.

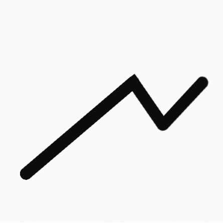
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final line.



click and dragging the start point.



Click multiple times to create multiple connected lines.



The polyline after confirming.

Arc Tool

Reference

Mode:: Draw Mode
Tool:: *Toolbar* ▸ *Arc*

The Arc tool create simple arcs using any of the *Draw* type brushes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions

The number of stroke points between each stroke edge.

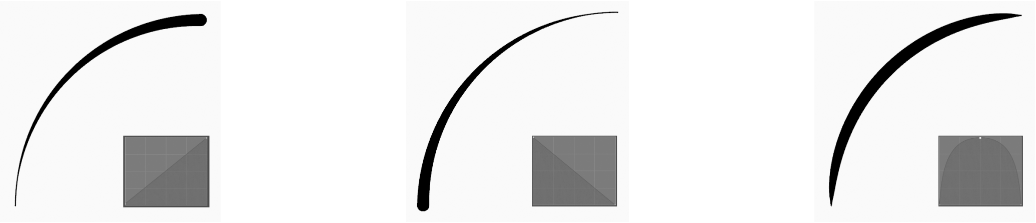
Thickness Profile

Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve

When enabled, the stroke use a curve profile to control the thickness along the arc.

Different thickness profile samples.



Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Color

Settings to determine the color of strokes.

See Color

Usage

Selecting a Brush and Material

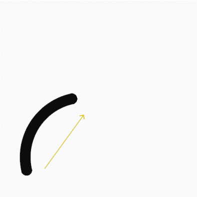
In the Tool Settings select the brush, material and color type to use with the tool. The Arc tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Arcs

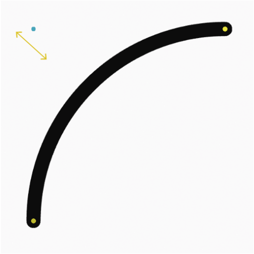
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can tweak the arc using a single cyan manipulator (hand icon).
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to make a perfect arc, use `Alt` to create the arc from a center point or `M` to flip.

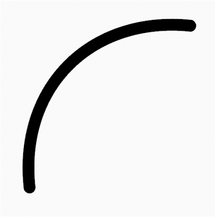
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final arc.



click and dragging the start point.



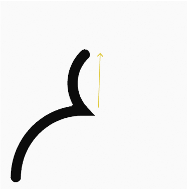
Tweaking arc with the manipulator.



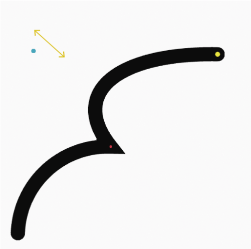
The arc after confirming.

Extruding

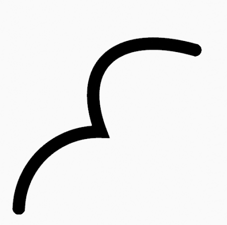
Before confirming you can use `E` to extrude the end point of the arc to generate multiple connected arcs.



End point extruding.



Tweaking the last arc with the manipulator.



The connected arcs after confirming.

Curve Tool

Reference

Mode:: Draw Mode
Tool:: *Toolbar ‣ Curve*

The Curve tool create complex Bézier style curves using any of the *Draw* type brushes..

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions

The number of stroke points between each stroke edge.

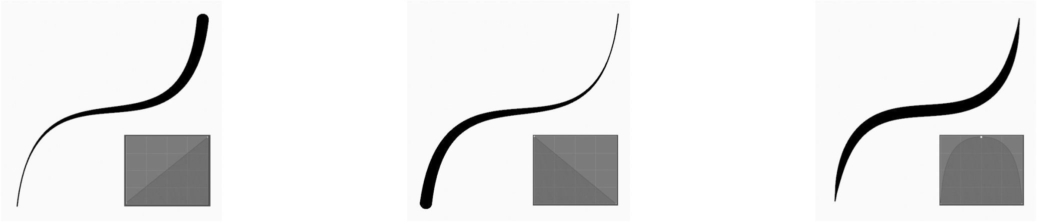
Thickness Profile

Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve

When enabled, the stroke use a curve profile to control the thickness along the curve.

Different thickness profile samples.



Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Color

Settings to determine the color of strokes.

See Color

Usage

Selecting a Brush and Material

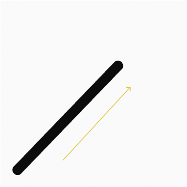
In the Tool Settings select the brush, material and color type to use with the tool. The Curve tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Curves

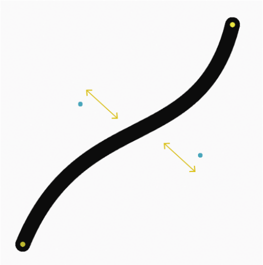
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can tweak the curve using two cyan Bézier like manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can hold `Shift` to use only one manipulator to tweak the curve (like the Arc tool), use `Alt` to create the arc from a center point.

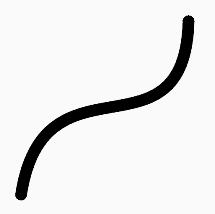
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final curve.



click and dragging the start point.



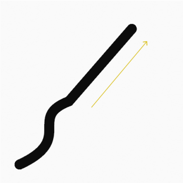
Tweaking curve with the manipulators.



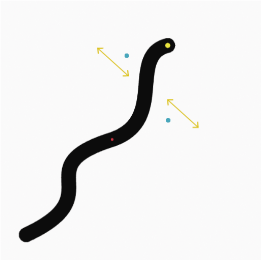
The curve after confirming.

Extruding

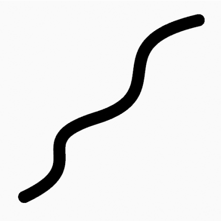
Before confirming you can use `E` to extrude the end point of the curve to generate multiple connected curves.



End point extruding.



Tweaking the last curve with the manipulators.



The connected curves after confirming.

Box Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ▸ *Box*

The Box tool create rectangular shapes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions

The number of stroke points between each stroke edge.

Thickness Profile

Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve

When enabled, the stroke use a curve profile to control the thickness along the line.

Usage

Selecting a Brush and Material

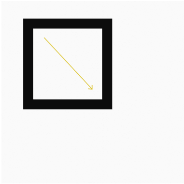
In the Tool Settings select the brush, material and color type to use with the tool. The Box tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Boxes

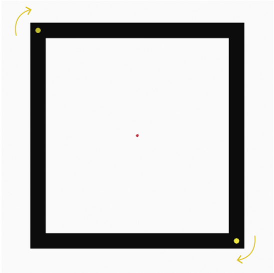
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to make a perfect square or use `Alt` to create the box from a center point.

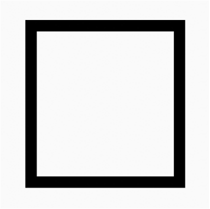
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final box.



click and dragging the start point.



Moving start and end points with manipulators.



The box after confirming.

Circle Tool

Reference

Mode::	Draw Mode
Tool::	*Toolbar* ‣ *Circle*

The Circle tool create oval shapes using any of the *Draw* type brushes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: Draw Brush.

Subdivisions
    The number of stroke points between each stroke edge.

Thickness Profile
    Use a curve widget to define the stroke thickness from the start (left) to end (right) of the stroke.

    Use Curve
        When enabled, the stroke use a curve profile to control the thickness along the line.

Brush Asset

Picks the brush asset used by the tool.

See Brush Asset for more information.

See Draw Brushes for a detailed list of all draw brushes and their options.

Brush Settings

Parameters to control to look of the stroke.

See Draw Brushes for details.

Color

Settings to determine the color of strokes.

See Color

Usage

Selecting a Brush and Material

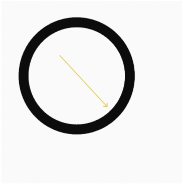
In the Tool Settings select the brush, material and color type to use with the tool. The Circle tool uses *Draw Brush* types. See Brush Settings for more information.

Creating Circles

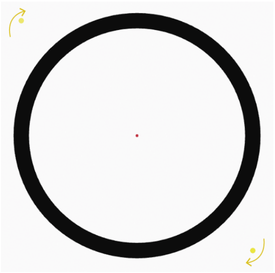
1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to make a perfect circle or use `Alt` to create the circle from a center point.

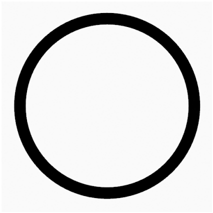
`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final circle.



Click and dragging the start point.



Moving start and end points with manipulators.



The circle after confirming.

Interpolate

Reference

Mode:: Draw Mode
Tool:: *Toolbar ▸ Interpolate*

The Interpolate tool interpolates strokes between the previous and next keyframe by adding a *single* keyframe. When you are on a frame between two keyframes and click and drag a new breakdown keyframe will be added. This way you define the final interpolation for the new stroke.

Usage

Set the Playhead on the Timeline between the two keyframes you want to interpolate. Click and drag from left to right to set the desired interpolation percentage and release to confirm, a new breakdown keyframe will be added.

Tool Settings

Layer
    Restrict the interpolation to Active or All layers.

Only Selected *Edit Mode*
    When enabled, only selected strokes will be interpolated.

Exclude Breakdowns
    Exclude existing Breakdowns keyframes as interpolation extremes.

Flip Mode
    Invert strokes start and end. Automatic will try to found the right mode for every stroke.

Smooth

Amount of smoothing to apply to interpolated strokes for reducing jitter/noise.

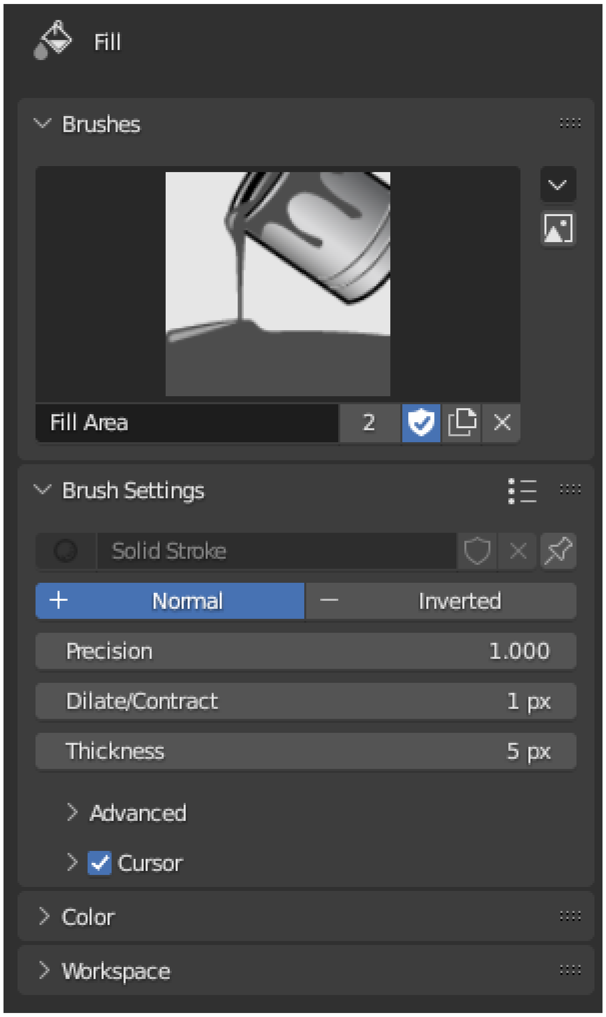
Iterations

Number of time to smooth newly created strokes.

Tools Settings

- Brush Asset
    - Brush Types
- Brush Settings
- Color
    - Palette

Brush Asset


Fill
Brushes
Fill Area
2
Brush Settings
Solid Stroke
Normal
Inverted
Advanced
Cursor
Color
Workspace

Precision 1.000
Dilate/Contract 1 px
Thickness 5 px

Brush data-block panel.

Brush

The Data-Block Menu to select a preset brush type or a custom brush.

Add Brush

When you add a brush, the new brush is a clone of the current one.

Brush Specials

Reset Brush

Reset the current brush to its default settings.

Reset All Brushes

Reset all brushes to their default settings.

Custom Icon

Allows definition of a custom brush icon.

Image Path

Defines the path to the image to use as custom icon.

Note

In order to save a custom brush in a blend-user, enable *Fake User*.

Brush Types

Draw Brushes

Draw brushes are the special type of brushes that uses Grease Pencil for drawing tools. The brush can be changed in the Tool Settings. The different draw brushes (pencil, Ink, marker, etc.) are settings variations of the same *Draw Brush*. You can create many brushes, each with unique settings to get different artistic result while drawing.

Fill Brushes

Fill brushes are the special type of brushes that uses Grease Pencil for the *Fill* tools. The brush can be changed in the Tool Settings. The different fill brushes are settings variations of the same *Fill Brush*. You can create many brushes, each with unique settings to get different result when filling areas.

Erase Brushes

Erase brushes are the special types of brushes that uses Grease Pencil for *Erase* tools. The brush can be changed in the Tool Settings. Soft and hard eraser brushes are settings variations of the same *Erase Brush*. You can create many brushes, each with unique settings to get different effects while erasing. The *Erase Brush* has also other two special eraser types: point and stroke.

Brush Settings

Material

Data-block selector for the material. Except for the *Erase* tool of course.

Pin Material (pin icon)

Pin the material to the brush.

The final appearance of the strokes is a combination of the brush and material used, binding the material to the brush gives more control and avoids a lack of coordination between the two.

Color

Paint Mode

Controls the source of the stroke color. The mode can be pinned to the brush by enabling the Pin icon in the Tool Settings header.

Material::	Use the stroke/fill base color material.
Color Attribute::	Use Color Attribute.

Color Picker

Sets the primary brush color.

Color, Secondary Color

The color of the brush. See Color Picker.

Mode

The color transformation will be applied on the stroke and/or the fill color.

Stroke::	Only paint over strokes.
Fill::	Only paint over fill areas.
Stroke & Fill::	Paint over strokes and fill areas.

Mix Factor

Mixing factor between the selected color and the base material color.

Palette

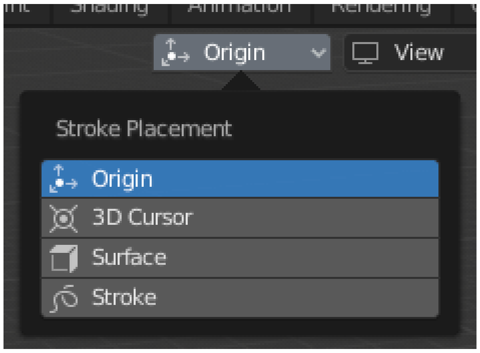
Active Color Palette. See Color Palette.

Stroke Placement

Reference

Mode:: Draw Mode
Header:: *Stroke Placement*

The Stroke Placement selector helps to select the location in which strokes are drawn.


Shading
Animation
Rendering
Origin
View
Stroke Placement
Origin
3D Cursor
Surface
Stroke

Stroke Placement pop-over.

Note

The Stroke Placement selection only affects new strokes and does not affect the existing ones.

Origin:: Strokes are placed at Grease Pencil object origin.

3D Cursor:: Strokes are placed at 3D cursor.

Surface:: Strokes will stick on mesh surfaces.

- Offset
  - Distance from the mesh surface to place the new strokes.
- Project Onto Selected
  - Only project the strokes onto selected objects.

Stroke:: Strokes will stick on other strokes.

- Target
  - **All Points::** All the points of the new stroke sticks to other strokes.
  - **End Points::**

	Only the start and end points of the new stroke sticks to other strokes.
First Point::	Only the start point of the new stroke sticks to other strokes.

Stroke using different Stroke Placements.

Origin.

3D Cursor.

Surface.

Stroke.

Drawing Plane

Reference

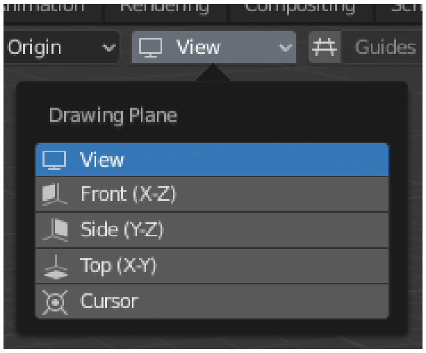
Mode::	Draw Mode and Sculpt Mode
Header::	*Drawing Plane*

The Drawing Planes selector helps to select the plane in which strokes are drawn.

To see which plane you are using when drawing strokes, you can enable *Canvas* in [Viewport Overlays](). See [Viewport Display]() to know more about Canvas settings.

Note

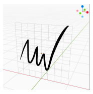
The Drawing Plane only affects new strokes and does not affect existing strokes.



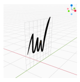
Drawing Planes pop-over.

View::	Strokes are drawn with the current 3D Viewport orientation.
Front (X-Z)::	Strokes are drawn on the plane determined by the XZ axes (front view).
Side (Y-Z)::	Strokes are drawn on the plane determined by the YZ axes (side view).
Top (X-Y)::	Strokes are drawn on the plane determined by the XY axes (top view).
Cursor::	Strokes are drawn with the current 3D cursor orientation.

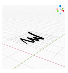
Stroke using different Drawing Planes with Canvas overlay activated.



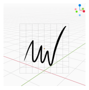
Front.



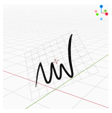
Side.



Top.



View.



Cursor.

Drawing Operations

Active Layer

Reference

Mode::	Draw Mode
Menu::	*Draw ‣ Active Layer*
Shortcut::	`Y`

Select the active layer.

Animation

Reference

Mode::	Draw Mode
Menu::	*Draw ‣ Animation*
Shortcut::	`I`

The stroke animation operations are described in the Animation section.

Interpolate Sequence

Reference

Mode::	Draw Mode
Menu::	*Draw ‣ Interpolate Sequence*

See Interpolate Sequence.

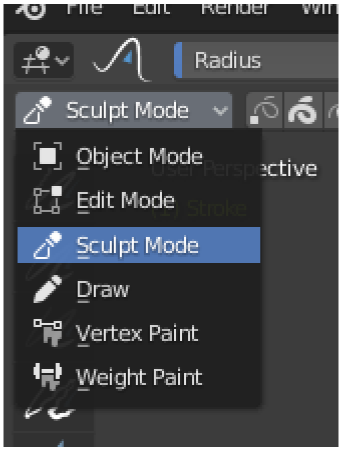
Sculpt Mode

- Introduction
  - Sculpt Mode
  - Sculpting Options
  - Keyboard Shortcuts
- Sculpting Brushes
- Sculpting Tools
- Brush Settings
  - Advanced
  - Cursor

Introduction

Sculpt Mode is similar to *Edit Mode* in that it is used to alter the shape of a drawing, but Sculpt Mode uses a very different workflow: Instead of dealing with individual elements (points and edit lines), an area of the model is altered using a brush. In other words, instead of selecting a group of points, Sculpt Mode manipulates the drawing in the brush region of influence.

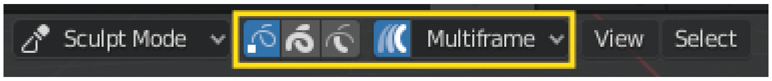
Sculpt Mode


File
Edit
Render
Radius
Sculpt Mode
Object Mode
Edit Mode
Sculpt Mode
Draw
Vertex Paint
Weight Paint

3D Viewport Mode selector: Sculpt Mode.

Sculpt Mode is selected from the *Mode* menu in the 3D Viewport header. Once Sculpt Mode is activated, the Toolbar of the 3D Viewport will change to Sculpt Mode specific panels. A red circle will appear and follow the location of the cursor in the 3D Viewport.

Sculpting Options


Sculpt Mode
Multiframe
View
Select

General sculpting options.

Selection Mask

Sculpt Mode in Grease Pencil allows you to select points or strokes to restrict the effect of the sculpting tools to only a certain areas of your drawing.

You can use the selection tools in the Toolbar for a quick selection. You can restrict sculpting only on the selected points or strokes with the Selection mode buttons. The three modes can be toggled with `1`, `2`, or `3` respectively.

Multiframe

Sometimes you may need to modify several frames at the same time with the sculpting tools.

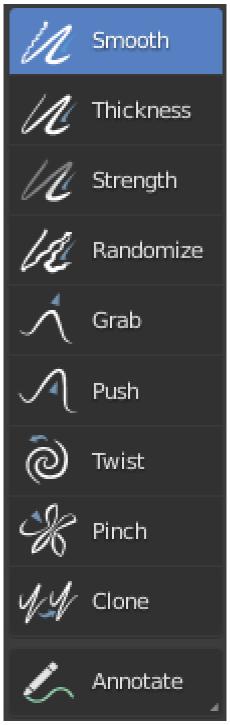
You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See Multiframe for more information.

Keyboard Shortcuts

- Invert stroke toggle `Ctrl`
- Change active material `U`

Sculpting Brushes

Brushes for Grease Pencil *Sculpt* mode bundled in the *Essentials* library. See Brush for more information.



Smooth Stroke

Eliminates irregularities in the area of the drawing within the brush's influence by smoothing the positions of the points.

Thickness Stroke

Increase or decrease the points thickness in the area of the drawing within the brush's influence.

Strength Stroke

Increase or decrease the points transparency (alpha) in the area of the drawing within the brush's influence.

Randomize Stroke

Add noise to the strokes in the area of the drawing within the brush's influence by moving points location in a random way.

Grab Stroke

Used to drag a group of points around. Unlike the other brushes, Grab does not modify different points as the brush is dragged across the model. Instead, Grab selects a group of points on mouse-down, and pulls them to follow the mouse. The effect is similar to moving a group of points in Edit Mode with Proportional Editing enabled.

Pull Stroke

Moves points in the direction of the brush stroke.

Twist Stroke

Twist the points in counter-clockwise (CCW) or Clockwise (CW) rotation.

Pinch Stroke

Pulls points towards the center of the brush. The inverse setting is Inflate, in which points are pushed away from the center of the brush.

Clone Stroke

Adds copies of the strokes in the clipboard in the center of the brush. You have to copy the selected strokes you want into the clipboard with `Ctrl-C` before using the tool.

Sculpting Tools

Brush
: Tool to use for any of the sculpting brushes.

Annotate
: Draw free-hand annotation.

  Annotate Line
  : Draw straight line annotation.

  Annotate Polygon
  : Draw a polygon annotation.

  Annotate Eraser
  : Erase previous drawn annotations.

Brush Settings

Reference

Mode:: Sculpt Mode
Panel:: *Sidebar ‣ Tool ‣ Brush Settings*

Radius

This option controls the radius of the brush, measured in pixels. `F` allows you to change the brush size interactively by dragging the mouse and then `LMB` (the texture of the brush should be visible inside the circle). Typing a number then enter while using `F` allows you to enter the size numerically.

(Size Pressure)

Adjusts the radius based on the stylus pressure when using a Graphics Tablet.

Strength

Controls how much each application of the brush affects the model. For example, higher values cause the *Randomize* brush to add noise to the strokes more quickly, and cause the *Smooth* brush to soften the strokes more quickly.

You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the brush and then `LMB`. You can enter the size numerically also while in `Shift-F` sizing.

(Strength Pressure)

Adjusts the strength based on the stylus pressure when using a Graphics Tablet.

Use Falloff

When enabled, use Strength falloff for the brush. Brush Strength decays with the distance from the center of the brush.

Sculpt Strokes

Filters the effect of the brush over the strokes. Applies to *Smooth* and *Randomize* brushes only.

Note

If there is no filter selected *Affect Position* is the default behavior.

Affect Position

Toggles the brush effect on the position of the stroke points.

Affect Strength

Toggles the brush effect on the strength (alpha) of the stroke points.

Affect Thickness

Toggles the brush effect on the thickness of the stroke points.

Affect UV

Toggles the brush effect on the UV rotation of the stroke points.

Direction

The influence direction of the brush. This can be Add or Subtract.

Advanced

Auto-Masking

Strokes

Affect only strokes under the cursor.

Material

Affect only the active material.

Layer

Affect only the active layer.

Cursor

The cursor can be disabled by toggling the checkbox in the *Cursor* header.

Color

Set the color of the brush ring. Depending on the current mode there will be options to set a single Color or a Color for Adding/Subtracting.

Edit Mode

- Introduction
  - Accessing Stroke Editing Tools
- Selecting
  - Select Mode
  - Select All/None/Invert
  - Select Random
  - Select Alternated
  - Select More/Less
  - Select Similar
  - Select Linked
  - Select First/Last
- Editing Tools
- Grease Pencil Menu
  - Transform
  - Mirror
  - Snap
  - Active Layer
  - Animation
  - Interpolate Sequence
  - Duplicate
  - Copy
  - Paste
  - Weights
  - Show/Hide
  - Separate
  - Clean Up
  - Delete
- Stroke Menu
  - Subdivide
  - Subdivide and Smooth
  - Simplify
  - Trim
  - Join

    - Move to Layer
    - Assign Material
    - Set as Active Material
    - Arrange
    - Close
    - Toggle Cyclic
    - Set Caps
    - Switch Direction
    - Set Uniform Thickness
    - Set Uniform Opacity
    - Scale Thickness
    - Set Curve Type
    - Set Curve Resolution
    - Reset UVs
- Point Menu
    - Extrude
    - Smooth
    - Vertex Groups
    - Set Handle Type

Introduction

Blender provides a variety of tools for editing Grease Pencil strokes. These are tools used to add, duplicate, move and delete elements.

Note

Editing multiple Grease Pencil objects at once is currently not supported.

Accessing Stroke Editing Tools

These are available through the different tools in the Toolbar, the Stroke menu in the 3D Viewport header, and context menus in the 3D Viewport, as well as individual shortcut keys.

Toolbar

When you select a stroke and `Tab` into Edit Mode, the Toolbar changes from *Object Tools* to *Stroke editing Tools*. These are only some of the stroke editing tools.

Menus

The Stroke Menu is located in the header.

Context Menu

`RMB` brings up the complete Stroke Menu.

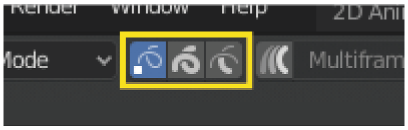
Selecting

Select Mode

Reference

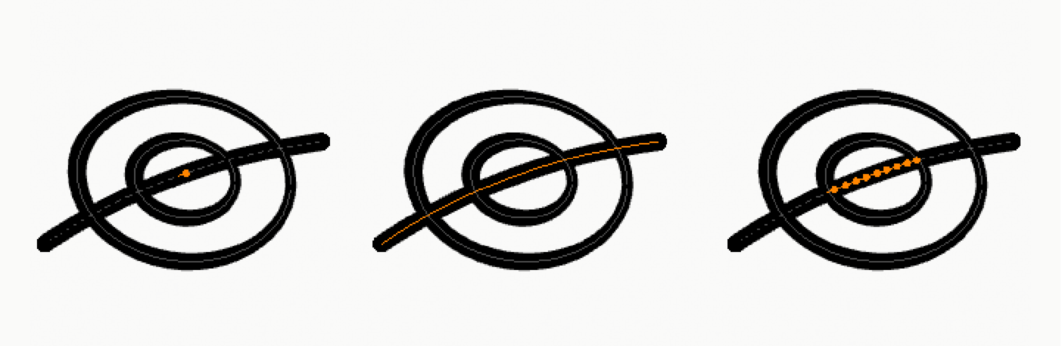
Mode:: Edit Mode
Menu:: *3D Viewport Header ‣ Select Mode*
Shortcut:: `1`, `2`, `3`

In Edit Mode there are three different selection modes. You can enter the different modes by selecting one of the three buttons in the header.



Edit Mode selection buttons.

Points::
To select individual points.
Strokes:: To select an entire stroke.
Segments:: To select all points that are between other strokes.



Points, stroke and in between stroke selection sample.

Select All/None/Invert

All these options have the same meaning and behavior as in Object Mode.

Select Random

Reference

Mode:: Edit Mode
Menu:: *Select ‣ Select Random*

Randomly selects unselected points or strokes.

Ratio
: The likelihood of an unselected elements being selected. Note that, this is not the percentage amount of elements that will be selected.

Random Seed
: Seed used by the pseudo-random number generator.

Action
: Selection or deselection of elements.

Select Alternated

Reference

Mode:: Edit Mode
Menu:: *Select ‣ Select Alternated*

Selects alternate points in the selected strokes.

Select More/Less

Reference

Mode:: Edit Mode
Menu:: *Select ‣ More/Less*
Shortcut:: `Ctrl-NumpadPlus`, `Ctrl-NumpadMinus`

The purpose of these operators is to reduce or enlarge the current selection within a stroke (i.e. they will never “go outside” of a stroke or “jump” to another stroke in the same object).

More
: For each selected point, select *all* its linked points (i.e. one or two…).

Less
: For each selected point, if *all* points linked to this point are selected, keep this one selected. Otherwise, deselect it.

Hint

When *all* points of a stroke are selected, nothing will happen (as for *Less*, all linked points are always selected, and of course, *More* cannot add any). Conversely, the same goes when no points are selected.

Select Similar

Reference

Mode::	Edit Mode
Menu::	*Select ‣ Select Similar*
Shortcut::	`Shift-G`

Select all strokes with similar characteristics.

Mode
: The characteristics to compare.

Layer::	Selects all the points/strokes with a similar layer index.
Material::	Selects all the points/strokes with a similar material index.
Vertex Color::	

	Selects all the points/strokes with a similar vertex color.
Radius::	Selects all the points/strokes with a similar stroke radius.
Opacity::	Selects all the points/strokes with a similar layer opacity

Threshold

How similar the selection must be.

Select Linked

Reference

Mode::	Edit Mode
Menu::	*Select ‣ Select Linked*
Shortcut::	`L`, `Ctrl-L`

`L` (or `Ctrl-L` for all) will add to the selection the cursor's nearest control point, and all the linked ones, i.e. all points belonging to the same stroke.

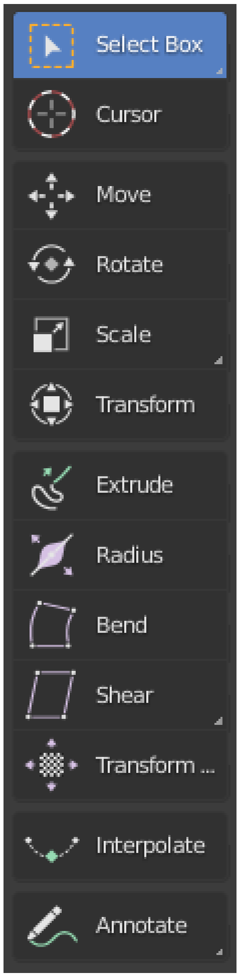
Select First/Last

Reference

Mode::	Edit Mode
Menu::	*Select ‣ First/Last*

These operators will toggle the selection of the first or last point(s) of the stroke(s) in the object. This is useful to quickly find the start of a stroke.

Editing Tools


Select Box
Cursor
Move
Rotate
Scale
Transform
Extrude
Radius
Bend
Shear
Transform ...
Interpolate
Annotate

Select

Select or moved.

Select Box

Select geometry by dragging a box.

Select Circle

Select geometry by painting on it.

Select Lasso

Select geometry by drawing a lasso.

Cursor

Change the location of the 3D Cursor.

Move

Translation tool.

Rotate

Rotation tool.

Scale

Scale tool.

Scale Cage

Change the scale of an object by controlling its cage.

Transform

Tool to adjust the objects translation, rotations and scale.

Extrude `E`

Extrusion tools duplicate points, while keeping the new geometry connected with the original points.

Radius `Alt-S`
: Expand or contract the thickness radius of the selected points.

Bend `Shift-W`
: Bend selected points between the 3D cursor and the pointer.

Shear `Shift-Ctrl-Alt-S`
: Shear selected points along the horizontal or vertical screen axis.

  To Sphere `Shift-Alt-S`
  : Move selected points outward in a spherical shape around the selected strokes' center.

Interpolate `Ctrl-E`
: Automatically create a breakdown keyframe between two normal keyframes.

Gradient
: Draw a line to set the fill material gradient for the selected strokes.

Annotate
: Draw free-hand annotation.

  Annotate Line
  : Draw straight line annotation.

  Annotate Polygon
  : Draw a polygon annotation.

  Annotate Eraser
  : Erase previous drawn annotations.

Grease Pencil Menu

Transform

Strokes can be edited by transforming the locations of points.

Move, Rotate & Scale

Reference

Mode::	Edit Mode
Tool::	*Toolbar ‣ Move, Rotate, Scale*
Menu::	*Grease Pencil ‣ Transform ‣ Move, Rotate, Scale*
Shortcut::	`G`, `R`, `S`

Like other elements in Blender, points and strokes can be moved `G`, rotated `R` or scaled `S` as described in the Basic Transformations section. When in *Edit Mode*, Proportional Editing is also available for the transformation actions.

Transform Snapping

Basic move, rotate and scale transformations for selected points/strokes. See Move, Rotate, Scale Basics for more information.

Tools

Reference

Mode::	Edit Mode
Menu::	*Grease Pencil ‣ Transform*
Tool::	*Toolbar ‣ Bend/Shear*

The *Bend*, *Shear*, *To Sphere*, *Extrude* and *Shrink Fatten* transform tools are described in the Editing tools section.

Mirror

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Mirror*
Shortcut:: `Ctrl-M`

The *Mirror* tool is also available, behaving exactly the same as with mesh vertices.

Snap

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Snap*
Shortcut:: `Shift-S`

Mesh snapping also works with Grease Pencil components.

Active Layer

Reference

Mode:: Edit Mode, Draw Mode
Menu:: *Grease Pencil ‣ Active Layer*
Shortcut:: `Y`

Select the active layer.

Animation

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Animation*
Shortcut:: `I`

The stroke animation operations are described in the Animation section.

Interpolate Sequence

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Interpolate Sequence*

See Interpolate Sequence.

Duplicate

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Duplicate*
Shortcut:: `Shift-D`

Duplicates the selected elements, without creating any connections with the rest of the strokes (unlike *Extrude*, for example), and places the duplicate at the location of the original elements.

Copy

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Copy*
Shortcut:: `Ctrl-C`

Copy the selected points/strokes to the clipboard.

Paste

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Paste*
Shortcut:: `Ctrl-V`

Paste Grease Pencil points or strokes from the internal clipboard to the active layer.

Paste on Back `Shift-Ctrl-V`
    Add pasted strokes behind all strokes.

Keep World Transform

Keep the world transform of strokes from the clipboard unchanged.

Weights

Reference

Mode:: Edit Mode, Draw Mode
Menu:: *Grease Pencil ‣ Weights*

The stroke weight operations are described in the Weights Menu section.

Show/Hide

Contains operators to adjust the visibility of points and strokes in the viewport.

Show All Layers

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Show/Hide ‣ Show All Layers*
Shortcut:: `Alt-H`

Shows all Grease Pencil layers.

Hide Active Layer

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Show/Hide ‣ Hide Active Layer*
Shortcut:: `H`

Hides the active Grease Pencil layers.

Hide Inactive Layers

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Show/Hide ‣ Hide Active Layer*

Shortcut:: `Shift-H`

Hides the all Grease Pencil layers except the active layer.

Separate

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Separate*
Shortcut:: `P`

Separate the selected geometry into a new Grease Pencil object.

Selected Points
: Separate the selected geometry into a new object.

Material
: Separates the geometry by each material.

Active Layer
: Separate all the strokes in the active layer into a new object. See 2D Layers for more information.

Clean Up

These tools help to cleanup degenerate geometry on the strokes.

Clean Loose Points

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Clean Up ‣ Delete Loose Points*

Removes strokes with only a few points.

Limit
: The number of points to consider a stroke as loose.

Delete Duplicate Frames

Reference

Mode::	Edit Mode
Menu::	*Grease Pencil ‣ Clean Up ‣ Delete Duplicate Frames*

Removes any duplicate keyframes.

Merge by Distance

Reference

Mode::	Edit Mode
Menu::	*Grease Pencil ‣ Clean Up ‣ Merge by Distance*

Simplifies a stroke by merging the selected points that are closer than a specified distance to each other. Note, unless using *Unselected*, selected points must be contiguous, else they will not be merged.

Merge Distance
  Sets the distance threshold for merging points.

Unselected
  Allows points in selection to be merged with unselected points. When disabled, selected points will only be merged with other selected ones.

Reproject

Reference

Mode::	Edit Mode
Menu::	*Grease Pencil ‣ Clean Up ‣ Reproject*

Sometimes you may have drawn strokes unintentionally in different locations in the 3D space but they look right from a certain plane or from the camera view. You can use Reproject to flatten all the selected strokes from a certain viewpoint.

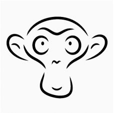
Reprojected Type

Front::	Reproject selected strokes onto the front plane (XZ).
Side::	Reproject selected strokes onto the side plane (YZ).
Top::	Reproject selected strokes onto the top plane (XY).
View::	Reproject selected strokes onto the current view.

Surface:: Reproject selected strokes onto the mesh surfaces.
Surface Offset
When Surface Mode is activated controls the stroke offset from the object.

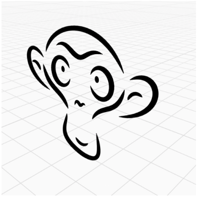
Cursor:: Reproject selected strokes onto 3D cursor rotation.

Keep Original
Maintains the original strokes after applying the tool.

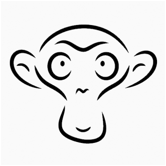


Original drawing from the front view.

Original drawing in the 3D Viewport.



Strokes reprojected onto the front plane to fix strokes misalignment.



Drawing after reprojection operation from the front view.

Delete

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Delete*
Shortcut:: `X`, `Delete`

Opens a pop-up menu with operators to remove geometry from the grease pencil object.

Frames
Deletes all the strokes at the current frame and in the current layer/channel.

Delete

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ‣ Delete ‣ Delete*

Deletes the selected points. When only one point remains, there is no more visible stroke, and when all points are deleted, the stroke itself is deleted.

Dissolve

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ▸ Delete ▸ Dissolve*
Shortcut:: `Ctrl-X`

Dissolving removes points between other points and connect the remaining points.

`Ctrl-X` Opens a pop-up to choose the dissolve type.

Dissolve

Deletes the selected points without splitting the stroke. The remaining points in the strokes stay connected.

Dissolve Between

Deletes all the points between the selected points without splitting the stroke. The remaining points in the strokes stay connected.

Dissolve Unselect

Deletes all the points that are not selected in the stroke without splitting the stroke. The remaining points in the strokes stay connected.

Delete Active Keyframe (Active Layer)

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ▸ Delete ▸ Delete Active Keyframe (Active Layer)*

Deletes all the strokes at the current frame in the active layer.

Delete Active Keyframes (All Layers)

Reference

Mode:: Edit Mode
Menu:: *Grease Pencil ▸ Delete ▸ Delete Active Keyframes (All Layers)*
Shortcut:: `Shift-Delete`

Deletes all the strokes at the current frame in all layer.

Stroke Menu

This page covers many of the tools in the *Strokes* menu. These are tools that work primarily on strokes, however, some also work with point selections.

Subdivide

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Subdivide*

Subdivides the strokes by inserting points between the selected points.

Number of Cuts
  The number of subdivisions to perform.

Selected Points
  When enabled, limits the effect to only the selected points within the stroke.

Subdivide and Smooth

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Subdivide and Smooth*

Subdivides and smooths the strokes by inserting points between the selected points.

Number of Cuts
  The number of subdivisions to perform.

Selected Points
When enabled, limits the effect to only the selected points within the stroke.

Iterations
Number of times to repeat the procedure.

Factor
The amount of the smoothness on subdivided points.

Smooth Endpoints
Smooths the stroke's endpoints.

Keep Shape
Preserves the strokes shape.

Position
When enabled, the operator affect the points location.

Radius
When enabled, the operator affect the points thickness.

Opacity
When enabled, the operator affect the points strength (alpha).

Simplify

Reference

Mode::	Edit Mode
Menu::	*Stroke ▸ Simplify*

Uses the RDP algorithm (Ramer-Douglas-Peucker algorithm) for points deletion. The algorithm tries to obtain a similar line shape with fewer points.

Factor

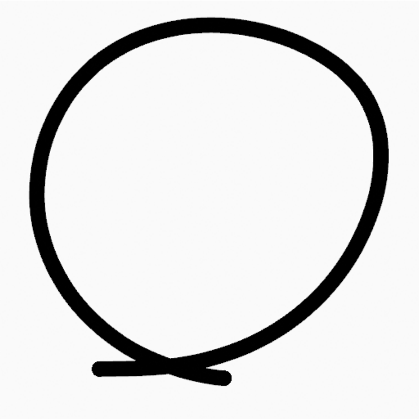
Controls the amount of recursively simplifications applied by the algorithm.

Trim

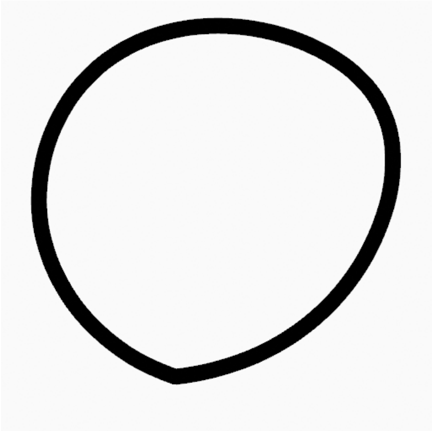
Reference

Mode::	Edit Mode
Menu::	*Stroke ▸ Trim*

Trims selected stroke to first loop or intersection.



Original stroke.



Result of trim operation.

Join

Join

Reference

Mode::	Edit Mode
Menu::	*Stroke ▸ Join ▸ Join,*

Join two or more strokes into a single one.

Type

Join::	`Ctrl-J` Join selected strokes by connecting points.
Join and Copy::	Join selected strokes by connecting points in a new stroke.

Leave Gaps

When enabled, do not use geometry to connect the strokes.

Join and Copy

Reference

Mode::	Edit Mode
Menu::	*Stroke* ‣ *Join* ‣ *Join and Copy*
Shortcut::	`Shift-Ctrl-J`

Same as Join but *Type* defaults to *Join and Copy*.

Move to Layer

Reference

Mode::	Edit Mode
Menu::	*Stroke* ‣ *Move to Layer*
Shortcut::	`M`

A pop-up menu to move the stroke to a different layer. You can choose the layer to move the selected strokes to from a list of layers of the current Grease Pencil object. You can also add a new layer to move the selected stroke to. When creating a new layer, there is another pop-up to type in the name of the new layer.

Assign Material

Reference

Mode:: Edit Mode
Menu:: *Stroke ▸ Assign Material*

Changes the material linked to the selected stroke. You can choose the name of the material to be used by the selected stroke from a list of materials of the current Grease Pencil object.

Set as Active Material

Reference

Mode:: Edit Mode
Menu:: *Stroke ▸ Set as Active Material*

Sets the active object material based on the selected stroke material.

Arrange

Reference

Mode:: Edit Mode
Menu:: *Stroke ▸ Arrange*

Change the drawing order of the strokes in the 2D layer.

Bring to Front
  Moves to the top the selected points/strokes.

Bring Forward
  Moves the selected points/strokes upper the next one in the drawing order.

Send Backward

Moves the selected points/strokes below the previous one in the drawing order.

Send to Back
Moves to the bottom the selected points/strokes.

Close

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Close*
Shortcut::	F

Close or open strokes by connecting the last and first point.

Type

Close All::	Close all open selected strokes.
Open All::	Open all closed selected strokes.
Toggle::	Close or Open selected strokes as required.

Toggle Cyclic

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Toggle Cyclic*

Toggles between an open stroke and closed stroke (cyclic).

Type

Close All::	Close all open selected strokes.
Open All::	Open all closed selected strokes.
Toggle::	Close or Open selected strokes as required.

Set Caps

Reference

Mode:: Edit Mode
Menu:: *Stroke ‣ Set Caps*

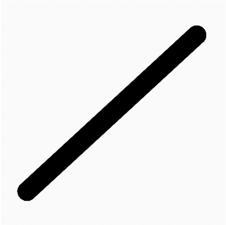
Toggle ending cap styles of the stroke.

Rounded
Sets stroke start and end points to rounded (default).

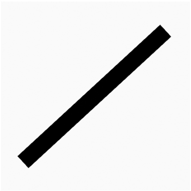
Flat
Toggle stroke start and end points caps to flat or rounded.

Toggle Start
Toggle stroke start point cap to flat or rounded.

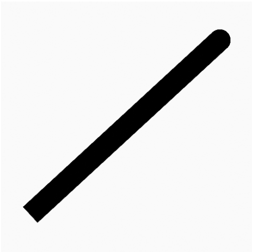
Toggle End
Toggle stroke end point cap to flat or rounded.



Stroke ending with rounded caps.



Stroke ending with flat caps.



Stroke ending with combined caps.

Switch Direction

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Switch Direction*

Reverse the direction of the points in the selected strokes (i.e. the start point will become the end one, and vice versa).

Set Uniform Thickness

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Set Uniform Thickness*

Makes the thickness equal for the entire stroke.

Thickness
  Thickness value to use on all points of the stroke.

Set Uniform Opacity

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Set Uniform Opacity*

Makes the opacity equal for the entire stroke.

Opacity
  Opacity value to use on all points of the stroke.

Scale Thickness

Reference

Mode::	Edit Mode

Menu::	*Stroke ‣ Scale Thickness*

When enabled, scales the stroke thickness during scale transformations.

Set Curve Type

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Set Curve Type*

Sets the spline type for the splines in the stroke component that are in the selection.

Type

The type to convert the splines in the selection to. Read the Spline Types page for more details on the different spline types.

Bézier::	Convert to a Bézier spline. A spline converted from a poly spline gets vector handles, while one converted from NURBS or Catmull Rom spline gets auto handles. **Note** When converting from a NURBS spline to a Bézier spline, at least six points are needed. When the number of points is not a multiple of three a full conversion is not possible and the spline has to be truncated.
NURBS::	Convert to a NURBS spline.
Poly::	Convert to a poly spline.
Catmull Rom::	Convert to a Catmull Rom spline.

Handles

Take handle information into account in the conversion

Set Curve Resolution

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Set Curve Resolution*

Sets the number of points generated along each curve segment (between two handles).

Reset UVs

Reference

Mode::	Edit Mode
Menu::	*Stroke ‣ Set Curve Resolution*

Reset UV transformation to default values.

Point Menu

Extrude

Reference

Mode::	Edit Mode
Menu::	*Point ‣ Extrude*
Tool::	*Toolbar ‣ Extrude*
Shortcut::	`E`

Extrudes points by duplicating the selected points, which then can be moved. The new points stay connected with the original points of the edit line.

Note

Since Grease Pencil strokes can only have one start an end point, a new stroke will be created when extrude intermediate points in the strokes.

Smooth

Reference

Mode::	Edit Mode
Menu::	*Point ‣ Smooth*

Softens strokes by reducing the differences in the locations of the points along the line, while trying to maintain similar values that make the line fluid and smoother.

Iterations

The number of times to repeat the procedure.

Factor
The amount of the smoothness to apply.

Smooth Endpoints
Smooths the stroke's endpoints.

Keep Shape
Preserves the strokes shape.

Position
When enabled, the operator affect the points location.

Radius
When enabled, the operator affect the points thickness.

Opacity
When enabled, the operator affect the points strength (alpha).

Vertex Groups

Operators for working with vertex groups.

Set Handle Type

Reference

Mode::	Edit Mode
Menu::	*Point ‣ Set Handle Type*
Shortcut::	V

Sets the handle type for the points on the Bézier curve that are in the selection.

Type
The handle type to switch to.

Free::	The handles are independent of each other.
Auto::	This handle has a completely automatic length and direction which is set by Blender to ensure the smoothest result. These handles convert to *Aligned* handles when moved.
Vector::	Both parts of a handle always point to the previous handle or the next handle which allows you to create curves or sections thereof made of straight lines or with sharp corners. Vector handles convert to *Free* handles when moved.
Aligned::	These handles always lie in a straight line, and give a continuous curve without sharp angles.

Vertex Paint Mode

- Introduction
    - Vertex Paint Mode
    - Vertex Paint Options
- Vertex Paint Brushes
- Vertex Paint Tools
- Brush Settings
    - Color Picker
    - Color Palette
    - Falloff
- Editing
    - Set Color Attribute
    - Reset Vertex Color
    - Invert
    - Levels
    - Hue/Saturation/Value
    - Brightness/Contrast

Introduction

Vertex Painting is a simple way of painting color onto a Grease Pencil object, by directly manipulating the color of points/vertices, rather than use only the materials base color.

Stroke with original base material color (left) and with vertex painting (right).

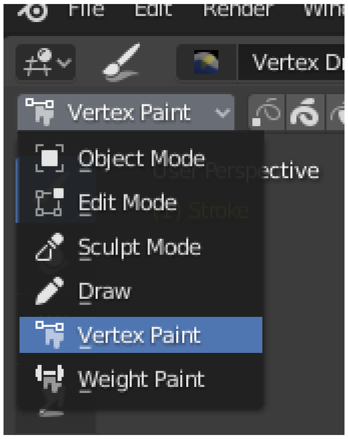
When a point is painted, the color of the points is mixing with the base material color according to the settings of the brush.

Note

A vertex in Grease Pencil is called point. Point and vertex names are equivalent.

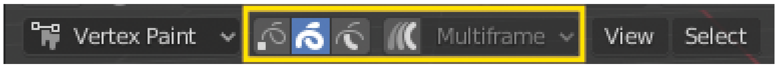
Vertex Paint Mode

Vertex Paint Mode is selected from the *Mode* menu in the 3D Viewport header. Once Vertex Paint Mode is activated, the Toolbar of the 3D Viewport will change to Vertex Paint Mode specific panels.


File
Edit
Render
Vertex Paint
Object Mode
Edit Mode
Sculpt Mode
Draw
Vertex Paint
Weight Paint

3D Viewport Mode selector set to Vertex Paint Mode.

Vertex Paint Options


Vertex Paint
Multiframe
View
Select

General Vertex Paint options.

Selection Mask

> Vertex Paint Mode in Grease Pencil allows you to select points or strokes to restrict the effect of the painting tools to only a certain areas of your drawing.
>
> You can use the selection tools in the Toolbar for a quick selections.
>
> You can restrict painting only on the selected points or strokes with the Selection mode toggle. The three modes can be toggled with `1`, `2`, or `3` respectively.

Multiframe

> Sometimes you may need to modify several frames at the same time with the painting tools.

You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See Multiframe for more information.

Vertex Paint Brushes

Brushes for Grease Pencil *Vertex Paint* mode bundled in the *Essentials* library.

Paint Point Color
: Paints a specified color over the object.

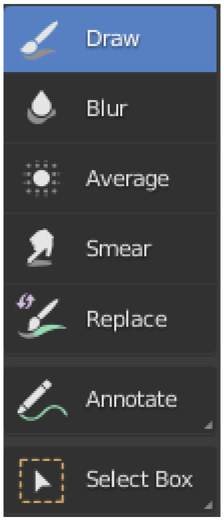
Blur Point Color
: Smooths out the colors of adjacent vertices. In this mode the Color Value is ignored. The strength defines how much the colors are blurred.

Average Point Color
: Smooths color by painting the average resulting color from all colors under the brush.

Smear Point Color
: Smudges colors by grabbing the colors under the brush and “dragging” them. This can be imagined as a finger painting tool.

Replace Point Color
: Change the color only to the stroke points that already have a color applied.

Vertex Paint Tools


Draw
Blur
Average
Smear
Replace
Annotate
Select Box

Brush
: Tool to use for any of the vertex paint brushes.

Annotate
: Draw free-hand annotation.

  Annotate Line
  : Draw straight line annotation.

  Annotate Polygon
  : Draw a polygon annotation.

  Annotate Eraser
  : Erase previous drawn annotations.

Brush Settings

Painting needs paint brushes and Blender provides a Brush panel within the Toolbar when in *Vertex Paint Mode*.

Brush

In the Data-Block menu you find predefined Brush presets. And you can create your own custom presets as needed.

Radius

This option controls the radius of the brush, measured in pixels. `F` allows you to change the brush size interactively by dragging the mouse and then `LMB` (the texture of the brush should be visible inside the circle). Typing a number then enter while using `F` allows you to enter the size numerically.

(Size Pressure)

Adjusts the brush radius based on the stylus pressure when using a Graphics Tablet.

Strength

How powerful the brush is when applied.

(Strength Pressure)

Adjusts the brush strength based on the stylus pressure when using a Graphics Tablet.

Mode

Stroke::	Only paint over strokes.
Fill::	Only paint over fill areas.
Stroke & Fill::	Paint over strokes and fill areas.

Cursor

See the global brush settings for Cursor settings.

Color Picker

The color of the brush. See Color Picker.

Note

Note that Vertex Paint works in sRGB space and the RGB representation of the same colors will be different between the paint tools and the materials that are in linear space.

Color Palette

The active Color Palette. See Color Palette.

Falloff

See the global brush settings for Falloff settings.

Editing

Set Color Attribute

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Set Color Attribute*

Sets the active color to all selected vertices.

Reset Vertex Color

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Reset Vertex Color*

Removes the Color Attribute information of the active strokes, if no strokes are selected, all strokes are reset.

Invert

Reference

Mode::	Vertex Paint Mode
Menu::	*Paint ‣ Invert*

Invert RGB values.

Levels

Reference

Mode:: Vertex Paint Mode
Menu:: *Paint ‣ Levels*

Adjust the levels of Color Attributes.

Hue/Saturation/Value

Reference

Mode:: Vertex Paint Mode
Menu:: *Paint ‣ Hue/Saturation/Value*

Adjust the color's HSV values.

Brightness/Contrast

Reference

Mode:: Vertex Paint Mode
Menu:: *Paint ‣ Brightness/Contrast*

Adjust the color's brightness/contrast.

Weight Paint Mode

- Introduction
  - Weight Paint
  - Weight Options
- Weight Paint Brushes
- Weight Paint Tools
- Tool Settings
  - Brush Settings
  - Options
- Weights Menu
  - Normalize All
  - Normalize
  - Invert
  - Smooth
  - Sample Weight

Introduction

Assigning weight to the points is primarily used for rigging strokes in cut-out animation, where the vertex groups are used to define the relative bone influences on the strokes. See Using Vertex Group for more information.

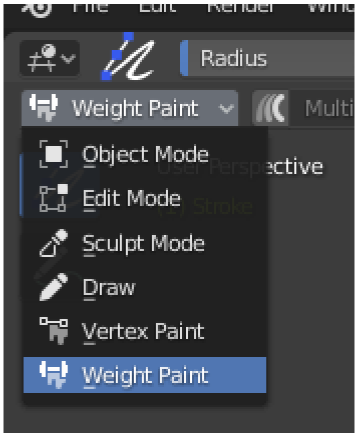
Note

A vertex in Grease Pencil is called point. Point and vertex names are equivalent.

Weight Painting is a method to maintain large amounts of weight information in an intuitive way. The selected Grease Pencil object is displayed slightly shaded with a rainbow color spectrum. The color visualizes the weights associated to each point in the active vertex group. By default blue means unweighted and red means fully weighted.

You assign weights to the points of the object by painting on it with weight brushes. Starting to paint on a strokes automatically adds weights to the active vertex group (a new vertex group is created if needed).

Weight Paint



3D Viewport Mode selector: Weight Paint Mode.

Weight Paint Mode is selected from the *Mode* menu in the 3D Viewport header. Once Weight Paint Mode is activated, the Toolbar of the 3D Viewport will change to Weight Paint Mode specific panels. A red circle will appear and follow the location of the cursor in the 3D Viewport.

Weight Options

Multiframe

> Sometimes you may need to assign weight to several frames at the same time with the Weight Paint tools.
>
> You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See Multiframe for more information.

Weight Paint Brushes

Brushes for Grease Pencil *Weight Paint* mode bundled in the *Essentials* library.

Paint Point Weight
: Paints a specified weight over the strokes.

Blur Point Weight
: Smooths out the weighting of adjacent points. In this mode the Weight Value is ignored. The strength defines how much the smoothing is applied.

Average Point Weight
: Smooths weights by painting the average resulting weight from all weights under the brush.

Smear Point Weight
: Smudges weights by grabbing the weights under the brush and “dragging” them. This can be imagined as a finger painting tool.

Weight Paint Tools

For Grease Pencil Weight Paint modes each brush type is exposed as a tool, the brush can be changed in the Tool Settings. See Brush for more information.

Weight
Annotate

Brush

Tool to use for any of the weight paint brushes.

Annotate

Draw free-hand annotation.

Annotate Line

Draw straight line annotation.

Annotate Polygon

Draw a polygon annotation.

Annotate Eraser

Erase previous drawn annotations.

Tool Settings

- Brush Settings
- Options

Brush Settings

Painting needs paint brushes and Blender provides a Brush panel within the Toolbar when in *Weight Paint Mode*.

Brush

  In the Data-Block menu you find predefined Brush presets. And you can create your own custom presets as needed.

Radius `F`

  The radius defines the area of influence of the brush.

  (Size Pressure)

    Adjusts the radius based on the stylus pressure when using a Graphics Tablet.

Strength

  This is the amount of paint to be applied per brush stroke.

  (Strength Pressure)

    Adjusts the strength based on the stylus pressure when using a Graphics Tablet.

Use Falloff

  When enabled, use Strength falloff for the brush. Brush Strength decays with the distance from the center of the brush.

Weight `Ctrl-F`

  The weight (visualized as a color) to be used by the brush.

  Using `Ctrl-RMB` you can set the weight to the value thats under the cursor.

Direction `D`

Brush direction toggle, *Add* adds weight value while *Subtract* removes weight value. This setting can be toggled with `D`.

Options

Auto Normalize

Ensures that all deforming vertex groups add up to one while painting. When this option is turned off, then all weights of a point can have any value between 0 and 1. However, when vertex groups are used as deform groups for character animation then Blender always interprets the weight values relative to each other. That is, Blender always does a normalization over all deform bones. Hence in practice it is not necessary to maintain a strict normalization and further normalizing weights should not affect animation at all.

This option works most intuitively when used to maintain normalization while painting on top of weights that are already normalized with another tool.

Weights Menu

Reference

Mode::	Edit Mode
Menu::	*Weights*

This page covers many of the tools in the *Weights* menu.

Normalize All

Reference

Mode::	Edit Mode
Menu::	*Weights ‣ Normalize All*

For each point, this tool makes sure that the sum of the weights across all vertex groups is equal to 1. It normalizes all of the vertex groups, except for locked groups, which keep their weight values untouched.

Lock Active
: Keep the values of the active group while normalizing all the others.

Normalize

Reference

Mode::	Edit Mode
Menu::	*Weights ‣ Normalize*

This tool only works on the active vertex group. All points keep their relative weights, but the entire set of weights is scaled up such that the highest weight value is 1.0.

Invert

Reference

Mode:: Edit Mode
Menu:: *Weights ‣ Invert*

Replaces each weight of the selected vertex group by × -1.0 weight.

Examples:

- Original 1.0 converts to 0.0
- Original 0.5 remains 0.5
- Original 0.0 converts to 1.0

Smooth

Reference

Mode:: Edit Mode
Menu:: *Weights ‣ Smooth*

Smooths the weights of the active vertex group.

Sample Weight

Reference

Mode:: Edit Mode
Menu:: *Weights ‣ Sample Weight*
Shortcut:: `Shift-X`

Adjust the Weight of the Draw tool to the weight of the vertex under the mouse cursor.

Object Mode

- Trace Image to Grease Pencil
    - Usage
    - Options

Trace Image to Grease Pencil

Reference

Mode::	Object Mode
Menu::	*Object ‣ Convert ‣ Trace Image to Grease Pencil*

The *Trace Image to Grease Pencil* tool traces a black and white image and generates Grease Pencil strokes. If the image is not black and white, it will be internally converted. For better results, convert the images manually to black and white. Also try to keep the resolution of the image small; high resolutions can produce very dense strokes.

Usage

1. Add an Image Empty to the scene.
2. Run *Trace Image to Grease Pencil*.

Options

Target Object

Determines if the image empty is kept or replaced.

New Object: Creates a new grease pencil object and keeps the image empty Selected Object: Replaces the image empty with the grease pencil object.

Radius

The thickness of the generated Grease Pencil strokes.

Color Threshold

Determine the Luminance threshold above which strokes are generated.

Turn Policy

Determines how to resolve ambiguities during decomposition of an image into paths.

Black::	Prioritizes to connect black (foreground) components.
White::	Prioritizes to connect white (background) components.
Left::	Always take a left turn.
Right::	Always take a right turn.
Minority::	Prioritizes to connect the color (black or white) that occurs least frequently in the local neighborhood of the current position.
Majority::	Prioritizes to connect the color (black or white) that occurs most frequently in the local neighborhood of the current position.
Random::	Choose pseudo-randomly.

Mode

Determines if the image being traced is a single image or image sequence.

Single::	The image empty is a single image or the current frame of an image sequence.
Sequence::	The image empty is an Image Sequence.

Start at Current Frame

When enabled, start the tracing process at the current image frame.

Trace Frame

Used to trace only one frame of the image sequence, set to zero to trace all.

Animation & Rigging

- Introduction
  - Animation
  - Rigging
- Keyframes
  - Introduction
  - Editing
  - Keying Sets
- Armatures
  - Introduction
  - Bones
  - Properties
  - Structure
  - Skinning
  - Posing
- Lattice
  - Editing
  - Properties
  - Usage
- Constraints
  - Introduction
  - Interface
  - Constraint Types
- Actions
  - Working with Actions
  - Properties
- Drivers
  - Introduction
  - Usage
  - Drivers Panel
  - Workflow & Examples
  - Troubleshooting
- Markers
  - Types

    - Visualization
    - Add Marker
    - Selecting
    - Editing
    - Bind Camera to Markers
- Shape Keys
    - Introduction
    - Shape Keys Panel
    - Workflow
- Motion Paths
    - Options
    - Example

Introduction

Animation

Animation is making an object move or change shape over time. Objects can be animated in many ways:

Moving as a whole object
  Changing their position, orientation or size in time;

Deforming them
  Animating their vertices or control points;

Inherited animation
  Causing the object to move based on the movement of another object (e.g. its parent, hook, armature, etc.).

In this chapter, we will cover the first two, but the basics given here are actually vital for understanding the following chapters as well.

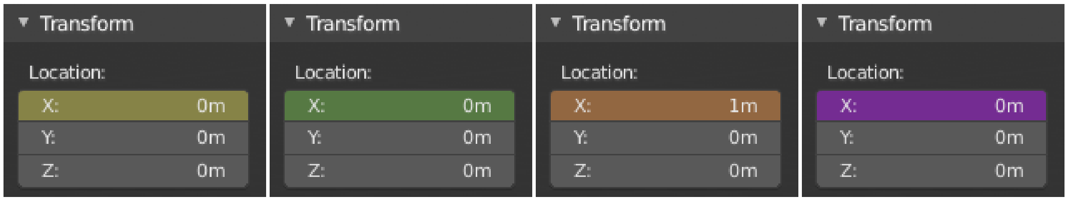
Animation is typically achieved with the use of keyframes.

See also

Related Sections

- Physical Simulation
- Motion Tracking

State Colors


Transform
Location:
X: 0m
Y: 0m
Z: 0m
Transform
Location:
X: 0m
Y: 0m
Z: 0m
Transform
Location:
X: 1m
Y: 0m
Z: 0m
Transform
Location:
X: 0m
Y: 0m
Z: 0m

State colors of properties.

Properties have different colors and menu items for different states.

Gray	Not animated
Yellow	Keyframed on the current frame
Green	Keyframed on a different frame
Orange	Changed from the keyframed value
Purple	Controlled by a driver

The changed value highlight currently doesn’t work with NLA.

Rigging

Rigging is a general term used for adding controls to objects, typically for the purpose of animation.

Rigging often involves using one or more of the following features:

Armatures

This allows mesh objects to have flexible joints and is often used for skeletal animation.

Constraints

To control the kinds of motions that make sense and add functionality to the rig.

Object Modifiers

Mesh deformation can be quite involved, there are multiple modifiers that help control this.

Shape Keys

To support different target shapes *(such as facial expressions)* to be controlled.

Drivers

So your rig can control many different values at once, as well as making some properties automatically update based on changes elsewhere.

Rigging can be as advanced as your project requires, rigs are effectively defining own user interface for the animator to use, without having to be concerned the underlying mechanisms.

Examples

- An armature is often used with a modifier to deform a mesh for character animation.
- A camera rig can be used instead of animating the camera object directly to simulate real-world camera rigs (with a boom arm, mounted on a rotating pedestal for example, effects such as camera jitter can be added too).

See also

The content of this chapter is simply a reference to how rigging is accomplished in Blender. It should be paired with additional resources such as Nathan Vegdahl’s excellent introduction to the fundamental concepts of character rigging, Humane Rigging.

Keyframes

- Introduction
    - Visualization
    - Interpolation
    - Keyframe Types
    - Handles & Interpolation Mode Display
- Editing
    - Insert Keyframe
    - Insert Keyframe with Keying Set
    - Delete Keyframes
    - Clear Keyframes
    - Editing Keyframes
    - Examples
- Keying Sets
    - Keying Set Panel
    - Adding Properties to a Keying Set
    - Set Active Keying Set
    - Whole Character Keying Set

Introduction

A *Keyframe* is simply a marker of time which stores the value of a property.

For example, a Keyframe might define that the horizontal position of a cube is at 3 m on frame 1.

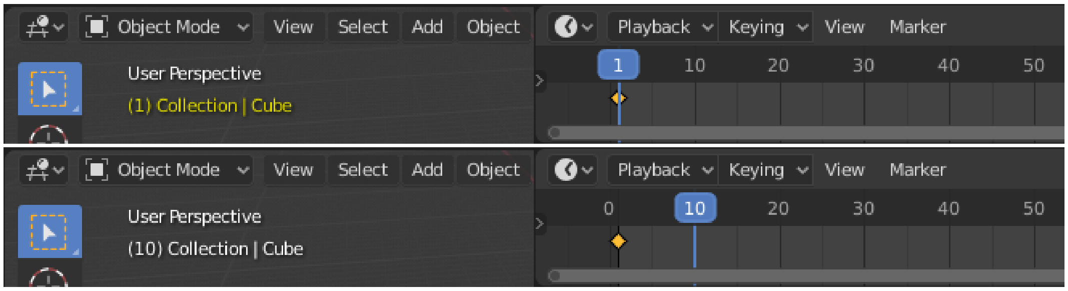
The purpose of a Keyframe is to allow for interpolated animation, meaning, for example, that the user could then add another key on frame 10, specifying the cube's horizontal position at 20 m, and Blender will automatically determine the correct position of the cube for all the frames between frame 1 and 10 depending on the chosen interpolation method (e.g. Linear, Bézier, Quadratic, etc.).

An overview of existing keyframes can be seen via the Dope Sheet editor.

Visualization

There are some important visualization features in the 3D Viewport that can help animation.

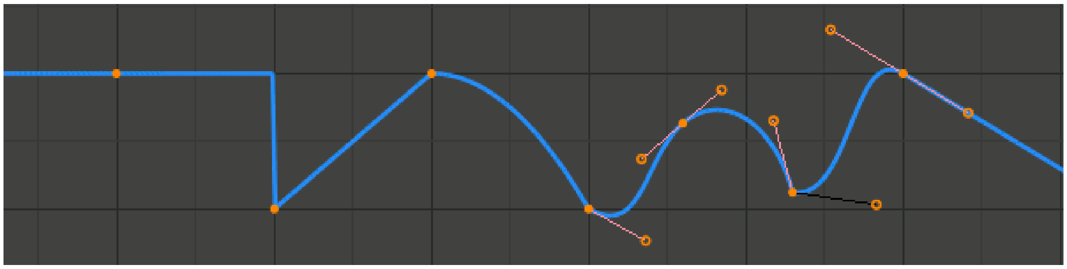
When the current frame is a keyframe for the current active object, the name of this object (shown in the upper left corner of the 3D Viewport) turns yellow.


Object Mode
View
Select
Add
Object
Playback
Keying
View
Marker
User Perspective
(1) Collection | Cube
1
10
20
30
40
50
Object Mode
View
Select
Add
Object
Playback
Keying
View
Marker
User Perspective
(10) Collection | Cube
0
10
20
30
40
50

Top: Current frame is a keyframe for Cube. Bottom: Current frame isn't a keyframe.

Interpolation

Keyframe interpolation is represented and controlled by animation curves, also known as F-Curves. These curves can be viewed and modified via the Graph Editor.



Constant, Linear, Quadratic and Bézier interpolation, with Linear extrapolation.

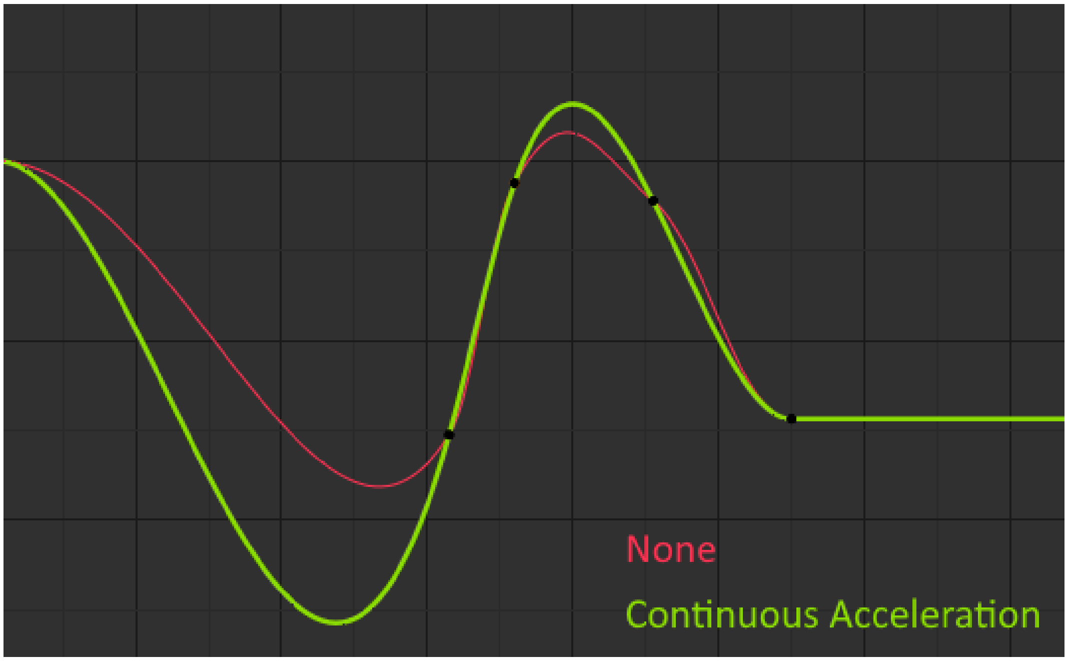
The X axis of the curve corresponds to time, while Y represents the value of the property. Keyframes themselves define points of the curve, while interpolation is controlled by additional parameters.

The Interpolation Mode is the main setting that specifies for each keyframe how the curve is interpolated from that key to the next one. There are a number of modes with fixed shapes, e.g. *Constant*, *Linear*, *Quadratic* etc, and a free form *Bézier* mode.

Extrapolation specifies how the curve extends before the first, and after the last keyframe. The main available choices are *Constant* and *Linear*; it is also possible to configure the curve to loop.

Bézier interpolation is controlled by handles, which have a handle type and position. The position of *Free* and *Aligned* handles must be set manually from the Graph editor, while *Vector*, *Automatic* and *Auto Clamped* handles are computed automatically from keyframe values.

Interpolation, Extrapolation and Handle Type can also be changed from the Dope Sheet editor.

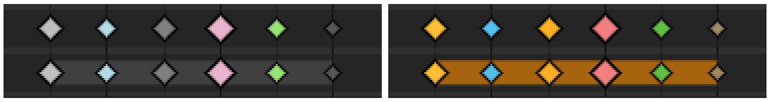


Handle smoothing modes. Yellow: *None*, Cyan: *Continuous Acceleration*.

The method how the three automatic handle types are computed is controlled by the per-curve Auto Handle Smoothing setting. The *None* mode resembles how most other software works and only considers the values of the immediately adjacent keys. The *Continuous Acceleration* mode considers the shape of the whole curve, which produces smoother results out of the box, but means that changes in one key affect interpolation over a larger section of the curve; it also tends to overshoot more with *Automatic* handles.

Keyframe Types

For visually distinguishing regular keyframes from different animation events or states (extremes, breakdowns, or other in-betweens) there is the possibility of applying different colors on them for visualization.



Keyframe (white / yellow diamond)
    Normal keyframe.

Left: not selected; Right: selected.

Breakdown (small cyan diamond)

Breakdown state. e.g. for transitions between key poses.

Moving Hold (dark gray / orange diamond)

A keyframe that adds a small amount of motion around a holding pose. In the Dope Sheet it will also display a bar between them.

Extreme (big pink diamond)

An ‘extreme’ state, or some other purpose as needed.

Jitter (tiny green diamond)

A filler or baked keyframe for keying on ones, or some other purpose as needed.

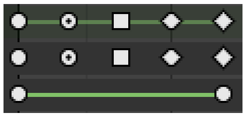
Generated (dark diamond)

A key generated by some tool, for example [Copy Global Transform: Fix to Camera](). This keyframe type indicates to Blender and add-ons that it is safe to remove and re-generate them, so be careful when manually marking your hand-made animation with this type.

Handles & Interpolation Mode Display

Dope Sheet can display the Bézier handle type associated with the keyframe, and mark segments with non-Bézier interpolation. This facilitates basic editing of interpolation without the use of the Graph Editor.

The icon shape represents the type of the [Bézier Handles]() belonging to the keyframe.



From top: summary, Bézier, linear.

Circle	Auto Clamped (default)
Circle With Dot	Automatic

Square	Vector
Clipped Diamond	Aligned
Diamond	Free

If the handles of a keyframe have different types, or in case of summary rows representing multiple curves, out of the available choices the icon that is furthest down the list is used. This means that if a grouped row uses a circle icon, it is guaranteed that none of the grouped channels have a non-auto key.

Horizontal green lines mark the use of non-Bézier Interpolation. The line is dimmed in summary rows if not all grouped channels have the same interpolation.

Display of this information can be disabled via the *Show Handles and Interpolation* option of the Dope Sheet’s View Menu.

Editing

Insert Keyframe

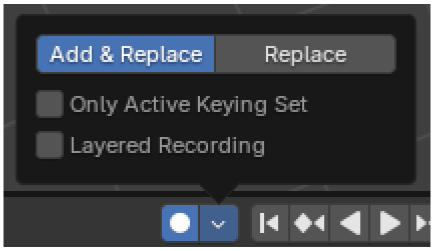
Reference

Mode:: Object Mode
Menu:: *Object ‣ Animation ‣ Insert Keyframe*
Shortcut:: `I`

There are several methods of adding new keys. Namely:

- In the 3D Viewport, pressing `I` will key properties based on the Default Key Channels User Preferences.
- When a Keying Set is active, it is used instead of reading the User Preferences.
- Hovering over a property and pressing `I` or with the context menu by `RMB` a property and choose *Insert Keyframe* from the menu.
- With the User Preference "Pie Menu on Drag" enabled, holding down `I` and moving the cursor will bring up a pie menu to insert one of Location, Rotation, Scale, and Available.

Auto Keyframe



Timeline Auto Keyframe.

Auto Keyframe is the record button in the *Timeline* header. Auto Keyframe adds keyframes automatically to the set frame if the value for transform type properties changes.

See Timeline Keyframe Control for more info.

Insert Keyframe with Keying Set

Reference

Mode::	Object Mode
Menu::	*Object* ‣ *Animation* ‣ *Insert Keyframe with Keying Set*
Shortcut::	`K`

Insert Keyframes for specified Keying Set, with menu of available Keying Sets.

Delete Keyframes

Reference

Mode::	Object Mode
Menu::	*Object* ‣ *Animation* ‣ *Delete Keyframes…*
Shortcut::	`Alt-I`

There are several methods of removing keyframes:

- In the 3D Viewport press `Alt-I` to remove keys from selected objects on the current frame.
- When the mouse is over a value, press `Alt-I`.
- `RMB` a value and choose *Delete Keyframe* from the menu.

Clear Keyframes

Reference

Mode::	Object Mode
Menu::	*Object ‣ Animation ‣ Clear Keyframes…*
Shortcut::	`Shift-Alt-I`

Removes all keyframes from the selected object.

Editing Keyframes

Keyframes can be edited in two editors. To do so go to either the Graph Editor or the Dope Sheet.

Examples

Keyframe Animation

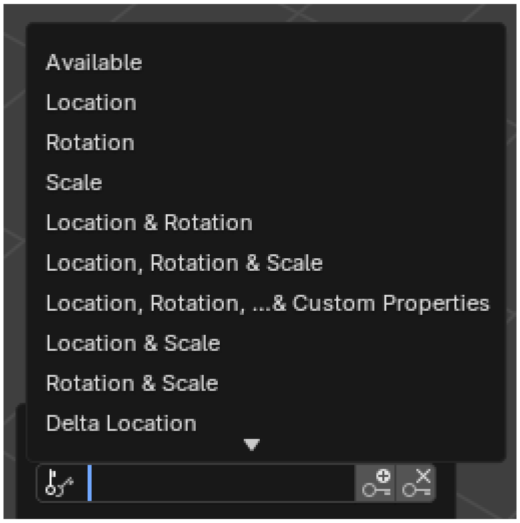
This example shows you how to animate a cube's location, rotation, and scale.

1. First, in the Timeline, or other animation editors, set the frame to 1.
2. With the cube selected in Object Mode, press `I` in the 3D Viewport. This will record the location, rotation, and scale, for the cube on frame 1.
3. Set the frame to 100.
4. Use Move `G`, Rotate `R`, Scale `S`, to transform the cube.
5. Press `I` in the 3D Viewport.

To test the animation, press `Spacebar` to play.

Keying Sets

Keying Sets are a collection of animated properties that are used to animate and keyframe multiple properties at the same time. For example, pressing `K` in the 3D Viewport will bring up the available Keying Sets. Blender will then add keyframes for whichever Keying Set is chosen. There are some built-in Keying Sets and also custom Keying Sets called “Absolute Keying Sets”.


Available
Location
Rotation
Scale
Location & Rotation
Location, Rotation & Scale
Location, Rotation, ...& Custom Properties
Location & Scale
Rotation & Scale
Delta Location

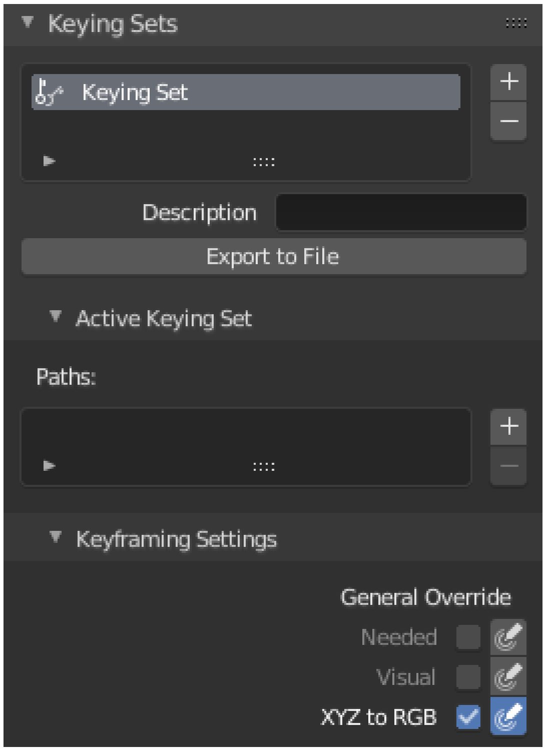
The Active Keying Sets data ID in the Timeline.

Keying Set Panel

Reference

Editor:: Properties
Panel:: *Scene* ▸ *Keying Set*

This panel is used to add, select, manage “Absolute Keying Sets”.


Keying Sets
Keying Set
Description
Export to File
Active Keying Set
Paths:
Keyframing Settings
General Override
Needed
Visual
XYZ to RGB

The Keying Set panel.

Active Keying Set

A List View of Keying Sets in the active scene. Selecting a keying set makes it active

Add +

Adds an empty Keying Set.

Remove –

Removes the active keying set.

Description

A short description of the Keying Set.

Export to File

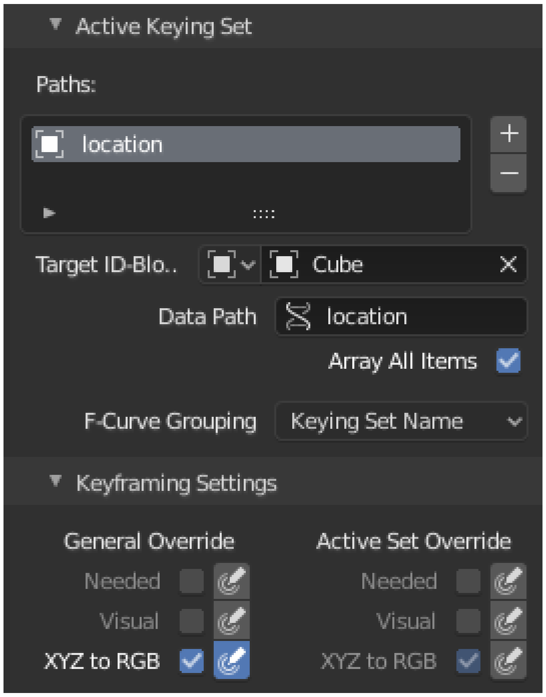
Export Keying Set to a Python script `File.py`. To re-add the Keying Set from the `File.py`, open then run the `File.py` from the Text Editor.

Active Keying Set Panel

Reference

Editor:: Properties
Panel:: *Scene ‣ Active Keying Set*

This panel is used to add properties to the active Keying Set.


Active Keying Set
Paths:
location
Target ID-Blo..
Cube
Data Path
location
Array All Items
F-Curve Grouping
Keying Set Name
Keyframing Settings
General Override
Active Set Override
Needed
Visual
XYZ to RGB
Needed
Visual
XYZ to RGB

The Active Keying Set panel.

Paths

A collection of paths in a List View each with a *Data Path* to a property to add to the active Keying Set.

Add +

Adds an empty path.

Remove –

Removes the selected path.

Target ID-Block

Set the ID Type and the *Object IDs* data path for the property.

Data Path
  Set the rest of the Data Path for the property.

Array All Items
  Use *All Items* from the Data Path or select the array index for a specific property.

F-Curve Grouping
  This controls what group to add the channels to.

  Keying Set Name, None, Named Group

Keyframing Settings

General Override
  These options control all properties in the Keying Set. Note that the same settings in *Preferences* override these settings if enabled.

Active Set Override
  These options control individual properties in the Keying Set.

Common Settings
  Needed
    Only insert keyframes where they are needed in the relevant F-Curves.

  Visual
    Insert keyframes based on the visual transformation.

Adding Properties to a Keying Set

Reference

Menu::	*Context menu ‣ Add All/Single to Keying Set*
Shortcut::	`K`

Some ways to add properties to Keying Sets.

`RMB` the property in the *User Interface*, then select *Add Single to Keying Set* or *Add All to Keying Set*. This will add the properties to the active Keying Set, or to a new Keying Set if none exist.

Hover the mouse over the properties, then press `K`, to add *Add All to Keying Set*.

Set Active Keying Set

Reference

Shortcut::	`Shift-K`

There are several ways to designate the active keying set:

- Press `Shift-K` in the 3D Viewport.
- Select a keying set in the Keying Set panel.
- Select a keying set in the Keying popover in the Timeline header,

Whole Character Keying Set

The built-in *Whole Character* Keying Set is made to keyframe all properties that are likely to get animated in a character rig. It was also implicitly used by the Old Pose Library system.

This keying set ignores bones whose name starts with one of the following prefixes, as it assumes these are technical bones that are not meant to be animated directly. The built-in Rigify addon generates such bones, for example.

- COR (Corrective)
- DEF (Deformation)
- GEO (Geometry)
- MCH (Mechanism)
- ORG (Original from meta rig)
- VIS (Visualization)

Armatures

- Introduction
    - Your First Armature
    - The Armature Object
- Bones
    - Introduction
    - Bone Collections
    - Structure
    - Tools
    - Selecting
    - Editing
    - Properties
- Properties
    - Introduction
    - Bone Collections
    - Selection Sets
    - Viewport Display
- Structure
    - Chains of Bones
- Skinning
    - Introduction
    - Armature Deform Parent
- Posing
    - Introduction
    - Selecting
    - Editing
    - Tool Settings
    - Bone Constraints

Introduction

An armature in Blender can be thought of as similar to the armature of a real skeleton, and just like a real skeleton an armature can consist of many bones. These bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way.

An “armature” is a type of object used for rigging. A rig is the controls and strings that move a marionette (puppet). Armature object borrows many ideas from real-world skeletons.

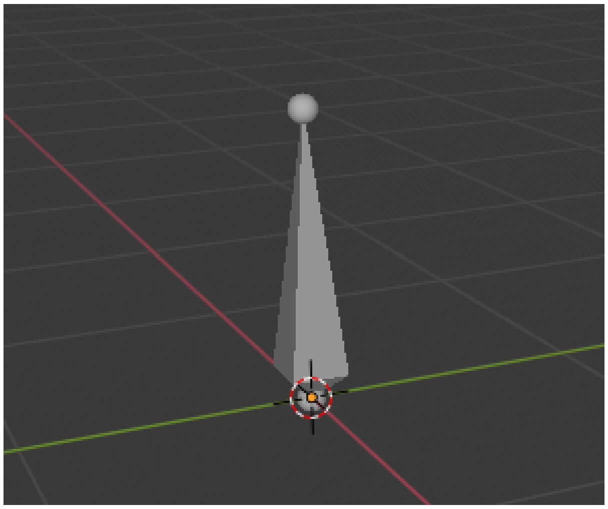
Your First Armature

In order to see what we are talking about, let us try to add the default armature in Blender.

(Note that armature editing details are explained in the armatures editing section.)

Open a default scene, then:

1. Delete all objects in the scene.
2. Make sure the cursor is in the world origin with `Shift-C`.
3. Press `Numpad1` to see the world in Front view.
4. Add a *Single Bone* (*Add* ▸ *Armature*).
5. Press `NumpadPeriod` to see the armature at maximum zoom.



The default armature.

The Armature Object

As you can see, an armature is like any other object type in Blender:

- It has an origin, a position, a rotation and a scale factor.
- It has an Object Data data-block, that can be edited in *Edit Mode*.
- It can be linked to other scenes, and the same armature data can be reused on multiple objects.
- All animation you do in *Object Mode* is only working on the whole object, not the armature's bones (use the *Pose Mode* to do this).

As armatures are designed to be posed, either for a static or animated scene, they have a specific state, called "rest position". This is the armature's default "shape", the default position/rotation/scale of its bones, as set in *Edit Mode*.

In *Edit Mode*, you will always see your armature in rest position, whereas in *Object Mode* and *Pose Mode*, you usually get the current "pose" of the armature (unless you enable the *Rest Position* button of the *Armature* panel).

Bones

- Introduction
  - Classification
- Bone Collections
  - Visibility
  - Library Overrides
  - Some history
- Structure
  - Roll
  - Bones Influence
- Tools
  - Toolbar
  - Tool Settings
- Selecting
  - Selecting Bone Joints
  - Selecting Bones
  - Select Mirror
  - More/Less
  - Select Linked
  - Select Similar
  - Select Pattern
  - Parent/Child
  - Extend Parent/Child
- Editing
  - Introduction
  - Transform
  - Bone Roll
  - Extrude
  - Duplicate
  - Fill Between Joints
  - Split
  - Separate Bones
  - Subdivide
  - Switch Direction

  - Symmetrize
  - Naming
  - Parenting
  - Properties
  - Delete
- Properties
  - Transform
  - Bendy Bones
  - Relations
  - Inverse Kinematics
  - Deform
  - Viewport Display
  - Custom Properties

Introduction

Bones are the base elements of armatures. The visualization of bones can be set in the Armatures Viewport Display.

Classification

Bones in an Armature can be generally classified into two different types:

1. Deforming Bones
2. Control Bones

Deforming Bones

Are bones which when transformed will result in vertices associated with them also transforming in a similar way. Deforming Bones are directly involved in altering the positions of vertices associated with their bones.

Control Bones

Are Bones which act in a similar way to switches, in that, they control how other bones or objects react when they are transformed. A Control Bone could for example act as a sliding switch control when the bone is in one position to the left, it could indicate to other bones that they react in a particular way when transformed, and when the Control Bone is positioned to the right, transforming other bones or objects could do something completely different. Control Bones are not directly used to alter the positions of vertices; in fact, Control Bones often have no vertices directly associated with themselves.

Bone Collections

Bone Collections group the bones of an Armature into named collections. The armature is the owner of these collections, so they are available in all modes. Bone Collections are identified by their name, which are unique within the Armature. Bone Collections can be nested inside other Bone Collections to create an organized hierarchy for complex rigs.

In the text below, “collection” is understood to refer to “bone collection”; Scene Collections are not described here.

Bone Collections can be managed via the Armature and Bone property panels.

Visibility

Bone Collections can be shown & hidden via the list in the Armature properties, as well as via the list in the Bone properties. Bone visibility is determined by the visibility of its collections, its own ‘solo’ and ‘hidden’ properties:

- If the bone itself is marked as ‘hidden’, it is invisible regardless of the bone collections.
- If a parent collection is hidden, child collections will also be hidden; same is true for soloed collections.
- A bone is visible when it is contained in any visible collection.
- If a collection is soloed, it will be visible regardless of the collection’s ‘hidden’ property.
- A bone that is not assigned to any bone collection is visible; otherwise it would be impossible to select it & assign it to a collection.

Library Overrides

Bone collections can be added using library overrides. For this to work, both the armature Object and the Armature itself need to be overridden.

Limitations

There are a few limitations when it comes to bone collections & overrides:

- Only bone collections that are local to the current blend file can be edited.
- Bone collections that already existed on the linked-in Armature are read-only, and only their visibility can be toggled. Those visibility changes won't be saved, though.
- Custom properties of overridden bone collections cannot be edited in the properties panel. Python access is fine; this is just a current limitation of Blender's UI code.

How It Works

Bone collections added via overrides are 'anchored' to the preceding collection, by name. Here is an example. The *italic* collections are defined on the linked Armature in `armature.blend`. The **bold** ones are added by overrides in `armature_shot_47.blend`.

- *FK Controls*
- *IK Controls*
- **Left Pinky** (anchored to "IK Controls")
- **Right Pinky** (anchored to "Left Pinky")

Now if the Armature in `armature.blend` gets updated with two more collections it might look like this:

- *FK Controls*
- *IK Controls*
- *Face Controls*
- *Face Detail Controls*

After reloading `armature_shot_47.blend`, it will look like this:

- *FK Controls*
- *IK Controls*
- **Left Pinky** (still anchored to “IK Controls”)
- **Right Pinky** (still anchored to “Left Pinky”)
- *Face Controls*
- *Face Detail Controls*

Some history

Bone Collections were introduced in Blender 4.0, as a replacement for armature layers and bone groups. Bone Collections are owned by the Armature, so they are available in all modes. To contrast, bone groups were stored on the object’s pose, and thus were not available in armature edit mode.

Structure

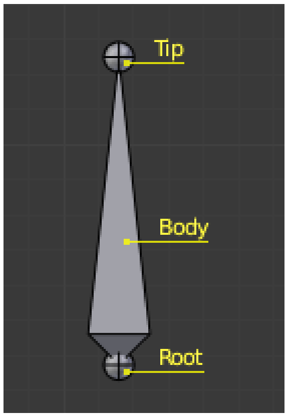
They have three elements:

1. The “start joint” named *root* or *head*.
2. The “body” itself.
3. And the “end joint” named *tip* or *tail*.

With the default armature in Edit Mode, you can select the root and the tip, and move them as you do with mesh vertices.

Both root and tip (the “joints”) define the bone by their respective position.

They also have a radius property, only useful for the envelope deformation method (see below).



The elements of a bone.

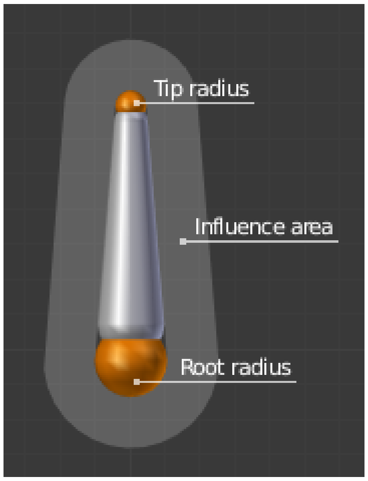
Roll

Activating the Axes checkbox will show local axes for each bone’s tip. The Y axis is always aligned along the bone, oriented from root to tip, this is the “roll” axis of the bones.

Bones Influence

Basically, a bone controls a geometry when vertices “follow” the bone. This is like how the muscles and skin of your finger follow your finger-bone when you move a finger.

To do this, you have to define the strength of *influences* a bone has on a certain vertex.

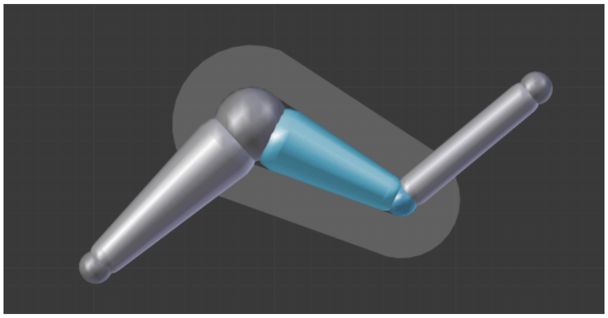


A bone in Envelope visualization, in Edit Mode.

The simplest way is to have each bone affecting those parts of the geometry that are within a given range from it. This is called the *envelope technique*, because each bone can control only the geometry "enveloped" by its own influence area.

If a bone is visualized as *Envelope*, in *Edit Mode* and in *Pose Mode* you can see the area of influence, which depends on:

- The *distance* property and
- the root's radius and the tip's radius.



Our armature in Envelope visualization, in Pose Mode.

All these influence parameters are further detailed in the skinning pages.

Tools

- Toolbar
- Tool Settings
    - Options

Toolbar

Mesh Edit Mode tools:

Select
    Select or move.

    Select Box
        Select geometry by dragging a box.

    Select Circle
        Select geometry by dragging a circle.

    Select Lasso
        Select geometry by drawing a lasso.

Cursor
    Change the location of the 3D Cursor.

Move
    Translation tool.

Rotate
    Rotation tool.

Scale
    Scale tool.

    Scale Cage
        Change the scale of an object by controlling its cage.

Transform
    Tool to adjust the objects translation, rotations and scale.

Annotate

Draw free-hand annotation.

Annotate Line

Draw straight line annotation.

Annotate Polygon

Draw a polygon annotation.

Annotate Eraser

Erase previous drawn annotations.

Measure

Measure distances in the scene.

Roll

Rotates a bone around its local Y axis.

Bone Size

Todo.

Bone Envelope

Todo.

Extrude

Creates a new bone connected to the last selected joint.

Extrude to Cursor

Creates a new bone between the last selected joint and the mouse position.

Shear

Todo.

Tool Settings

Options

X-Axis Mirror

See X-Axis Mirror Pose Mode.

Selecting

You can select and edit bones of armatures in *Edit Mode* and in *Pose Mode*. Here, we will see how to select bones in *Edit Mode*. Selecting bones in *Pose Mode* is similar to selecting in *Edit Mode* with a few specific differences that will be detailed in the posing part.

Similar to vertex/edge selection in meshes, there are two ways to select whole bones in *Edit Mode*:

1. Directly, by selecting the bone's body.
2. Selecting both of its joints (root and tip).

This is an important point to understand, because selecting bones' joints only might lead to non-obvious behavior, with respect to which bone you actually select.

Note that unlike the mesh display type, the armature display type has no effect on selection behavior. In other words, you can select a bone's joint or body the same way regardless of the bone visualization chosen.

Selecting Bone Joints

To select bones' joints you have the standard selection methods.

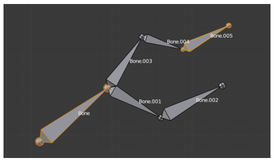
Inverse Selection

As stated above, you have to remember that these selection tools are for bones' joints only, not the bones' bodies.

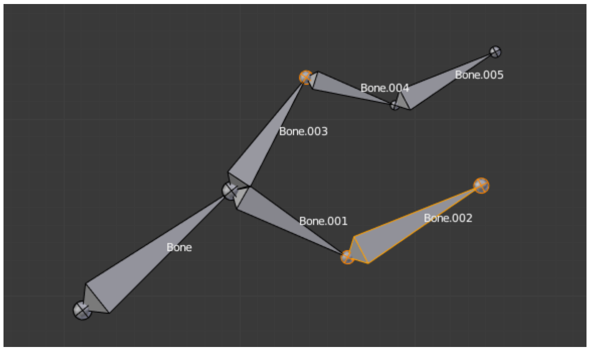
For example, the *Inverse* selection option `Ctrl-I` inverts the selection of bones' joints, not of bones (see *Inverse selection*).

Remember that a bone is selected only if both its joints are selected. So, when the selection status of bones' joints is inverted, a new set of bones is selected.

Inverse selection.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

Two bones selected.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

The result of the inverse selection `Ctrl-I`: The bones joints selection has been inverted, and not the bones selection.

Selecting Connected Bone Joints

Another example is: when you select the root of a bone connected to its parent, you also implicitly select the tip of its parent (and vice versa).

Note

Remember that when selecting bones' joints, the tip of the parent bone is the "same thing" as the root of its children bones.

Selecting Bones

By clicking on a bone's body, you will select it (and hence you will implicitly select its root and tip).

Using `Shift`-click, you can add to/remove from the selection.

You also have some *advanced selection* options, based on their relations.

Pick Shortest Path `Ctrl`-click
    Selects the path from the active bone to the bone under the mouse.

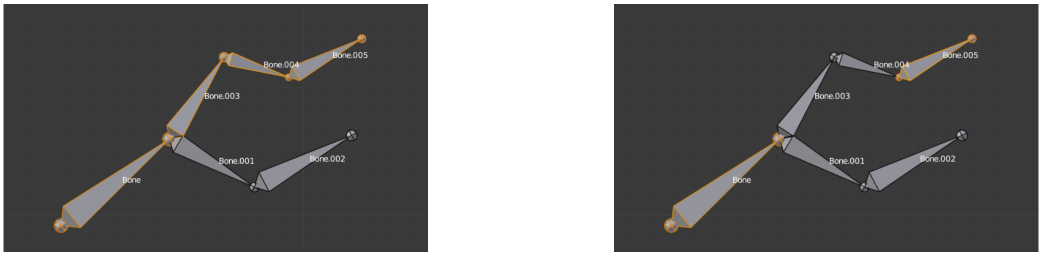
Deselecting Connected Bones

There is a subtlety regarding connected bones.

When you have several connected bones selected, if you deselect one bone, its tip will be deselected, but not its root, if it is also the tip of another selected bone.

To understand this, look at Fig. Bone deselection in a selected chain..

Bone deselection in a selected chain.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone
Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

A selected chain.

Two selected bones.

After `Shift`-clicking "Bone.003":

- "Bone.003" 's tip (which is same as "Bone.004" 's root) is deselected.
- "Bone" is "Bone.003" 's parent. Therefore, "Bone.003" 's root is the same as the tip of "Bone". Since "Bone" is still selected, its tip is selected. Thus the root of "Bone.003" remains selected.

Select Mirror

Reference

Mode:: Edit Mode
Menu:: *Select ‣ Select Mirror*
Shortcut:: `Shift-Ctrl-M`

Flip the selection from one side to another.

More/Less

Reference

Mode:: Edit Mode
Menu:: *Select ‣ More/Less*

More `Ctrl-NumpadPlus`
Expand the current selection to the connected bones.

Less `Ctrl-NumpadMinus`
Contrast the selection, deselect bones at the boundaries of each selection region.

Select Linked

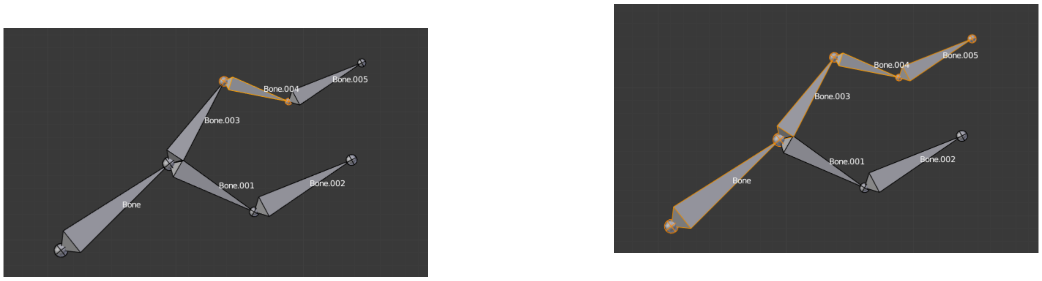
Reference

Mode:: Edit Mode
Menu:: *Select ‣ Select Linked*
Shortcut:: `Ctrl-L`

Selects all the bones in the chain which the active (last selected) bone belongs to.

All Forks
Selects all bones connected to the active bone even if the branch off from the current bone.

Linked bones selection.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone
Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

A single selected bone.

Its whole chain selected with Linked.

Select Simiar

Reference

Mode::	Edit Mode
Menu::	*Select* ▸ *Select Similar*
Shortcut::	`Shift-G`

Children
: Extends the selection to all hierarchical descendant bones.

Immediate Children
: Extends the selection to all direct child bones.

Siblings
: Selects bones that have the same parent as the active bone.

Length
: Selects bones with a similar bone length under the specified *Threshold*.

Direction (Y axis)
: Select bones aligned on the Y axis (along the bone’s length).

Prefix

Select bones with matching name prefix (separated by `.`).

Suffix
Select bones with matching name suffix (separated by `.`).

Bone Collection
Select bones that share one or more bone collections with the active bone.

Color
Select bones that have the same color as the active bone.

Shape
Select bones using the same shape object (in Pose Mode).

Select Pattern

Reference

Mode::	Pose & Armature Edit Modes
Menu::	*Select ‣ Select Pattern…*

Select bones by names, see Object Select Pattern for details.

Parent/Child

Parent `[`, Child `]`
You can deselect the active bone and select its immediate parent or one of its children.

Extend Parent/Child

Extend Parent `Shift-[`, Extend Child `Shift-]`
Similar to *Parent/Child* but it keeps the active bone in the selection.

Editing

- Introduction
    - Add Menu
    - Locking Bones
- Transform
    - Scale Radius
    - Scale Envelope Distance
    - Align Bones
- Bone Roll
    - Recalculate Roll
    - Set Roll
- Extrude
    - Mouse Clicks
- Duplicate
- Fill Between Joints
- Split
- Separate Bones
- Subdivide
- Switch Direction
- Symmetrize
- Naming
    - Naming Conventions
    - Auto-Name
    - Flip Names
- Parenting
    - Bone Collections
- Properties
- Delete
    - Bones
    - Dissolve

Introduction

As with any other object, you edit your armature in *Edit Mode* `Tab`.

The set of bone editing tools is quite similar to the one for mesh editing.

Important

One important thing to understand about armature editing is that you edit the *rest position* of your armature, i.e. its “default state”. An armature in its *rest position* has all bones with *no* rotation and scaled to 1.0 in their own local space.

The different poses you might create afterwards are based on this rest position. So if you modify it in *Edit Mode*, all the poses already existing will also be modified. Thus you should in general be sure that your armature is definitive before starting to skin and pose it!

Note

Please note that some tools work on bones’ joints, while others work on bones themselves. Be careful not to get confused.

Add Menu

Reference

Mode::	Edit Mode
Menu::	*Add*
Shortcut::	`Shift-A`

In the 3D Viewport, `Shift-A` to add a new bone to your armature.

This bone will be:

- Of one unit of length.
- Oriented towards the global Z axis.
- With its root placed at the 3D cursor position.
- With no relationship with any other bone of the armature.

Locking Bones

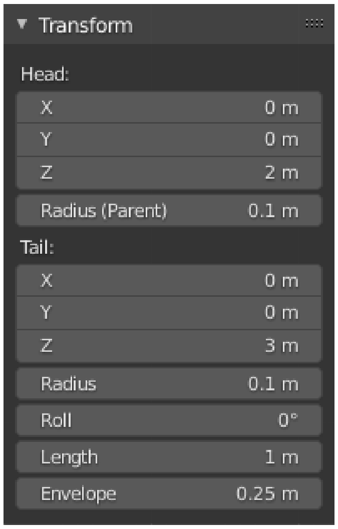
You can prevent a bone from being transformed in *Edit Mode* in several ways:

- All bones can be locked clicking on the *Lock* checkbox of their Transform panel in the *Bones* tab;
- Press `Shift-W` *Toggle Bone Options* ‣ *Locked*
- Select *Armature* ‣ *Bone Settings* ‣ *Toggle a Setting*.

If the root of a locked bone is connected to the tip of an unlocked bone, it will not be locked, i.e. you will be able to move it to your liking. This means that in a chain of connected bones, when you lock one bone, you only really lock its tip. With unconnected bones, the locking is effective on both joints of the bone.

Transform

We will not detail here the various transformations of bones, nor things like axis locking, pivot points, and so on, as they are common to most object editing, and already described in the mesh section. The same goes for mirroring, as it is nearly the same as with mesh editing. Just keep in mind that bones' roots and tips behave more or less like meshes' vertices, and bones themselves act like edges in a mesh.


Transform
Head:
X 0 m
Y 0 m
Z 2 m
Radius (Parent) 0.1 m
Tail:
X 0 m
Y 0 m
Z 3 m
Radius 0.1 m
Roll 0°
Length 1 m
Envelope 0.25 m

The Transform panel for armatures in Edit Mode.

As you know, bones can have two types of relationships: They can be parented, and in addition connected. Parented bones behave in *Edit Mode* exactly as if they had no relations. They can be moved, rotated, scaled, etc. without affecting their descendants. However, connected bones must always have parent's tips connected to child's roots, so by transforming a bone, you will affect all its connected parent/children/siblings.

While with other transform tools, the "local axes" means the object's axes, here they are the bone's own axes (when you lock to a local axis, by pressing the relevant key twice, the constraint is applied along the selected bone's local axis, not the armature object's axis).

Finally, you can edit in the *Transform* panel in the Sidebar region the positions and radius of both joints of the active selected bone, as well as its roll rotation.

Scale Radius

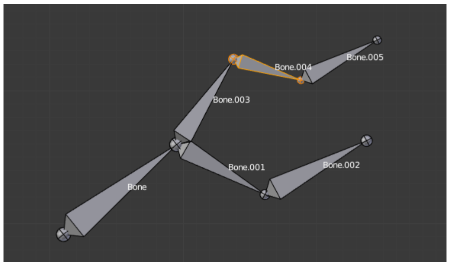
Reference

Mode:: Edit Mode
Menu:: *Armature ‣ Transform ‣ Scale Radius*
Shortcut:: `Alt-S`

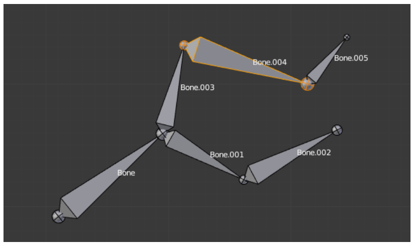
You can alter the radius that a bone has by selecting the head, body or tail of a bone, and then press `Alt-S` and move the mouse left or right. If the body is selected the mean radius will be scaled. And as usual, with connected bones, you scale at the same time the radius of the parent's tip and of the children's roots.

You can also alter the bone radius by selecting the tail or head of the bone you wish to alter, then navigate to *Properties ‣ Bone ‣ Deform ‣ Radius Section* and entering new values for the *Tail* and *Head* number fields.

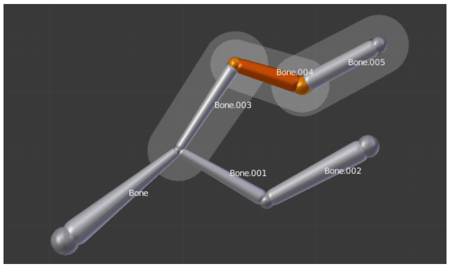
Bone Scale and Scale Radius comparison.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

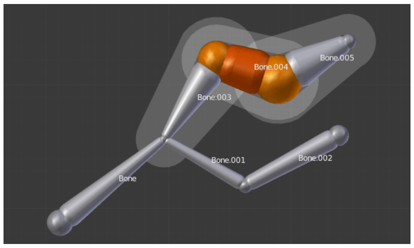
A single selected bone in Octahedron visualization.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

After normal scale.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

A single selected bone in Envelope visualization.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

After Scaled Radius. Its length remains the same, but its joints' radius are bigger.

Note that, when you resize a bone (either by directly scaling it, or by moving one of its joints), Blender automatically adjusts the end-radii of its envelope proportionally to the size of the modification. Therefore, it is advisable to place all the bones first, and only then edit their properties.

Scale Envelope Distance

Reference

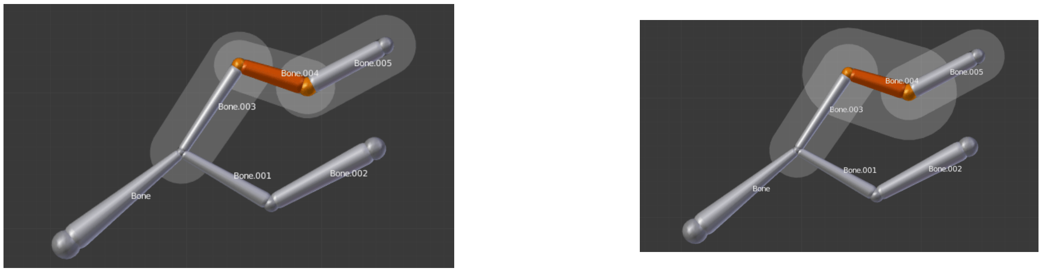
Mode::	Edit Mode and Pose Mode
Menu::	*Armature ‣ Transform ‣ Scale Envelope Distance*
Shortcut::	`Ctrl-Alt-S`

You can alter the size of the Bone Envelope volume by clicking on the body of the bone you want to alter, `Ctrl-Alt-S` then drag your mouse left or right and the Bone Envelope volume will alter accordingly.

You can also alter the Bone Envelope volume by selecting the Bone you wish to alter and then navigate to *Properties ‣ Bone ‣ Deform ‣ Envelope ‣ Distance* then enter a new value into it.

Altering the Bone Envelope volume does not alter the size of the bone just the range within which it can influence vertices of child objects.

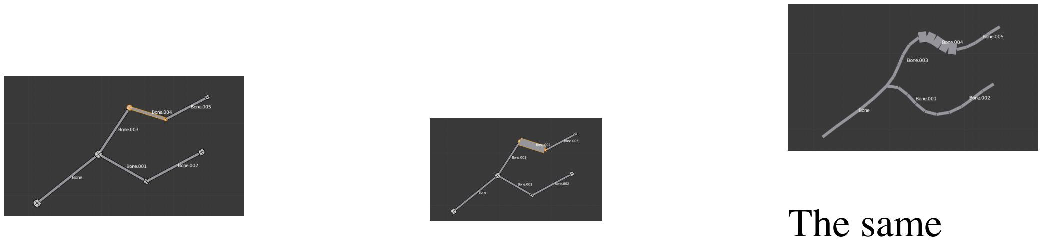
Envelope scaling example.



A single bone selected in Envelope visualization.

Its envelope distance scaled.

"Bone size" scaling example.


Bone.004
Bone.005
Bone.003
Bone
Bone.001
Bone.002

A single "default size" bone selected in B-Bone visualization.

Its envelope distance scaled.

The same armature in Object Mode and B-Bone visualization, with Bone.004's size scaled up.

Align Bones

Reference

Mode:: Edit Mode
Menu:: *Armature ‣ Transform ‣ Align Bones*
Shortcut:: `Ctrl-Alt-A`

Rotates the selected bones to achieve the same orientation as the active one.

Bone Roll

In *Edit Mode*, you can control the bone roll (i.e. the rotation around the Y axis of the bone).

However, after editing the armature, or when using Euler Rotation, you may want to set the bone roll.

Recalculate Roll

Reference

Mode::	Edit Mode
Menu::	*Armature* ‣ *Bone Roll* ‣ *Recalculate Roll*
Shortcut::	`Shift-N`

Axis Orientation

Local Tangent

Align roll relative to the axis defined by the bone and its parent.

X, Z

Global Axis

Align roll to global X, Y, Z axis.

X, Y, Z

Active Bone

Follow the rotation of the active bone.

View Axis

Set the roll to align with the viewport.

Cursor

Set the roll towards the 3D cursor.

Flip Axis
Reverse the axis direction.

Shortest Rotation
Avoids rolling the bone over 90 degrees from its current value.

Set Roll

Reference

Mode::	Edit Mode
Menu::	*Armature* ‣ *Bone Roll* ‣ *Set Roll*
Shortcut::	`Ctrl-R`

This is a transform mode where you can edit the roll of all selected bones.

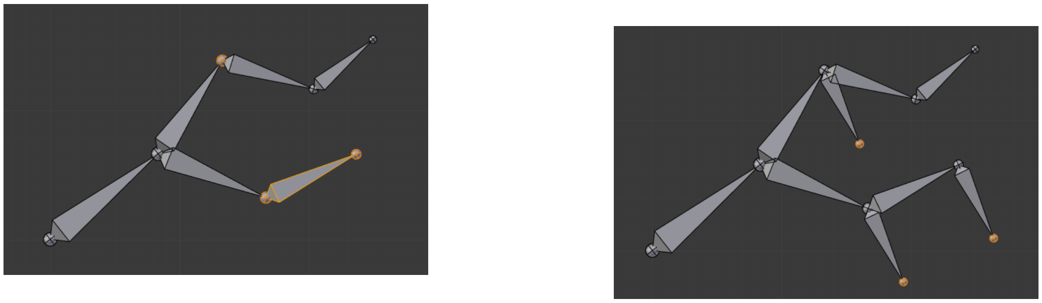
Extrude

Reference

Mode:: Edit Mode
Menu:: *Armature ‣ Extrude*
Shortcut:: `E`, `Shift-E`

When you press `E`, for each selected tip (either explicitly or implicitly), a new bone is created. This bone will be the child of "its" tip owner, and connected to it. As usual, once extrusion is done, only the new bones' tips are selected, and in select mode, so you can place them to your liking. See Fig. Extrusion example..

Extrusion example.

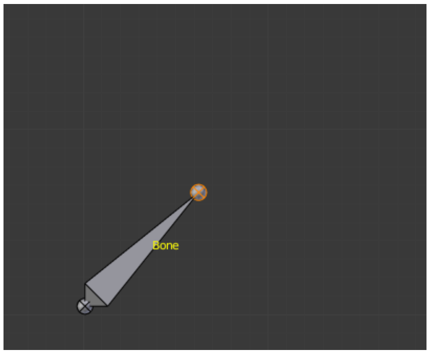


An armature with three selected tips.

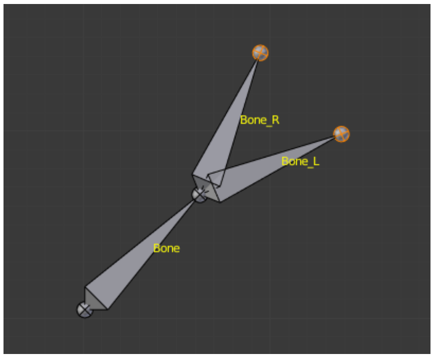
The three extruded bones.

You also can use the rotating/scaling extrusions, as with meshes, by pressing respectively `E R` and `E S` – as well as locked extrusion along a global or local axis.

Mirror extrusion example.



A single selected bone's tip.



The two mirror-extruded bones.

Bones have an extra "mirror extruding" tool, called by pressing `Shift-E`. By default, it behaves exactly like the standard extrusion. But once you have enabled X-Axis Mirror editing option, each extruded tip will produce *two new bones*, having the same name except for the "_L"/ "_R" suffix (for left/right, see the naming conventions). The "_L" bone behaves like the single one produced by the default extrusion – you can move, rotate or scale it exactly the same way. The "_R" bone is its mirror counterpart (along the armature's local X axis), see Fig. Mirror extrusion example..

Important

Canceling the extrude action causes the newly created bones to snap back to the source position, (creating zero length bones). These will be removed when exiting Edit Mode, however, they can cause confusion and it's unlikely you want to keep them. If you realize the problem immediately, undo the extrude action.

In case you are wondering, you cannot just press `X` to solve this as you would in mesh editing, because extrusion selects the newly created tips, and as explained below the Delete tool ignores bones' joints. To get rid of these extruded bones without undoing, you would have to move the tips, then select the bones and delete them.

Mouse Clicks

Reference

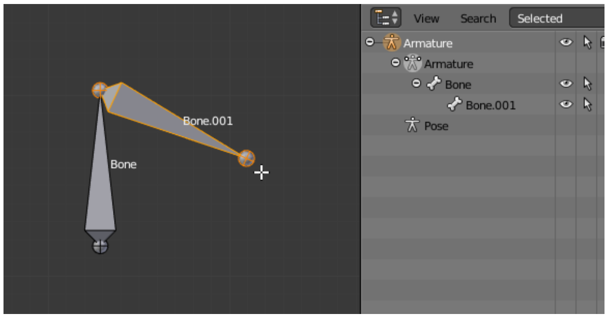
Mode::	Edit Mode
Shortcut::	`Ctrl-RMB`

If at least one bone is selected, `Ctrl-RMB`-clicking adds a new bone.

About the new bone’s tip:

After you `Ctrl-RMB`-clicked it becomes the active element in the armature, it appears to be right where you clicked, but (as in mesh editing) it will be on the plane parallel to the view and passing through the 3D cursor.

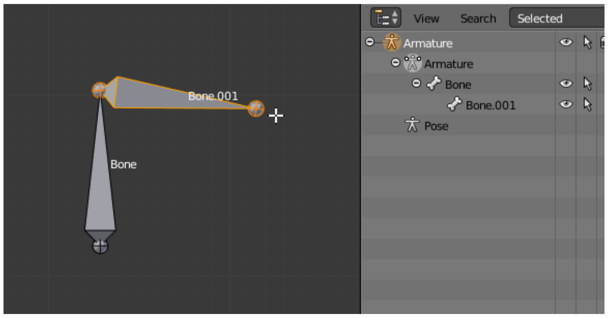
The position of the root and the parenting of the new bone depends on the active element:



`Ctrl`-clicking when the active element is a bone.

If the active element is a *bone*:

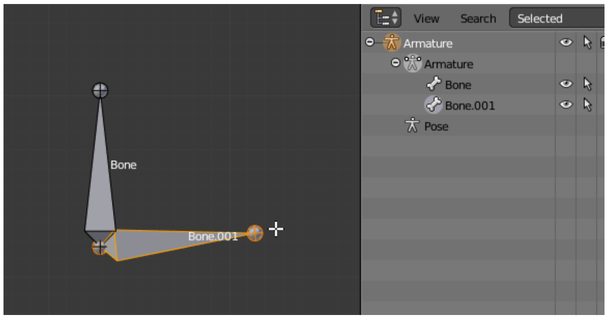
- The new bone’s root is placed on the active bone’s tip.
- The new bone is parented and connected to the active bone (check the Outliner in Fig. Ctrl-clicking when the active element is a tip.).


Bone.001
Bone
View
Search
Selected
Armature
Armature
Bone
Bone.001
Pose

`Ctrl`-clicking when the active element is a tip.

If the active element is a *tip*:

- The new bone's root is placed on the active tip.
- The new bone is parented and connected to the bone owning the active tip (check the Outliner in Fig. Ctrl-clicking when the active element is a tip.).


Bone
Bone.001
View
Search
Selected
Armature
Armature
Bone
Bone.001
Pose

`Ctrl`-clicking when the active element is a disconnected root.

If the active element is a *disconnected root*:

- The new bone's root is placed on the active root.
- The new bone is **not** parented to the bone owning the active root (check the Outliner in Fig. Ctrl-clicking when the active element is a disconnected root.).

And hence the new bone will **not** be connected to any bone.

Duplicate

Reference

Mode:: Edit Mode
Menu:: *Armature* ▸ *Duplicate*
Shortcut:: `Shift-D`

Note

This tool works on selected bones; selected joints are ignored.

As in mesh editing, by pressing `Shift-D` the selected bones will be duplicated. The duplicates become the selected elements and they are placed in select mode, so you can move them wherever you like.

If you select part of a chain, by duplicating it you will get a copy of the selected chain, so the copied bones are interconnected exactly like the original ones.

The duplicate of a bone which is parented to another bone will also be parented to the same bone, even if the root bone is not selected for the duplication. Be aware, though, that if a bone is parented **and** connected to an unselected bone, its copy will be parented, but **not** connected to the unselected bone (see Fig. Duplication example.).

Duplication example.

Fill Between Joints

Reference

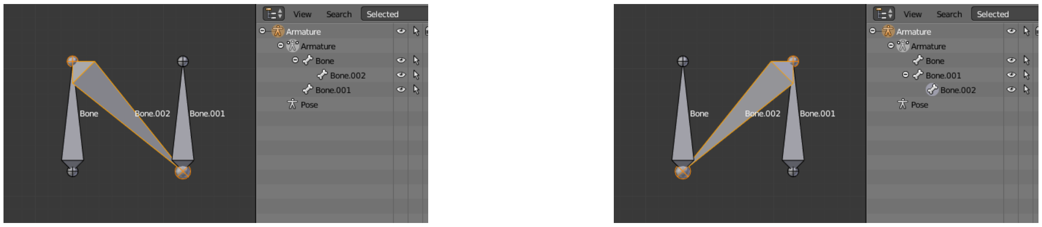
Mode:: Edit Mode
Menu:: *Armature ‣ Fill Between Joints*
Shortcut:: `F`

The main use of this tool is to create one bone between two selected joints by pressing `F`, similar to how in mesh editing you can “create edges/faces”.

If you have one root and one tip selected, the new bone:

- Will have the root placed on the selected tip.
- Will have the tip placed on the selected root.
- Will be parented and connected to the bone owning the selected tip.

Fill between a tip and a root.


View
Search
Selected
Armature
Armature
Bone
Bone.002
Bone.001
Pose
Bone
Bone.002
Bone.001
View
Search
Selected
Armature
Armature
Bone
Bone.001
Bone.002
Pose
Bone
Bone.002
Bone.001

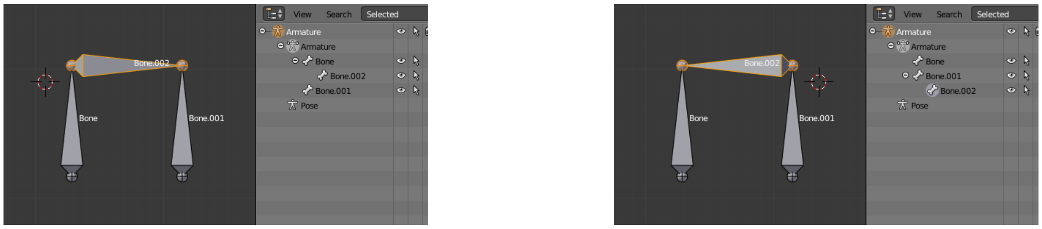
Active tip on the left.

Active tip on the right.

If you have two tips selected, the new bone:

- Will have the root placed on the selected tip closest to the 3D cursor.
- Will have the tip placed on the other selected tip.
- Will be parented and connected to the bone owning the tip used as the new bone’s root.

Fill between tips.


View
Search
Selected
Armature
Armature
Bone
Bone.002
Bone.001
Pose
Bone.002
Bone
Bone.001

3D cursor on the left.

3D cursor on the right.

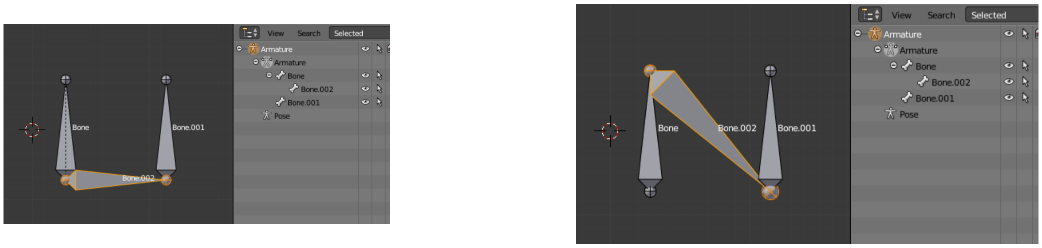
If you have two roots selected, you will face a small problem due to the event system in Blender not updating the interface in real-time.

When clicking `F`, similar to the previous case, you will see a new bone:

- With the root placed on the selected root closest to the 3D cursor.
- With the tip placed on the other selected root.
- Parented and connected to the bone owning the root used as the new bone's root.

If you try to move the new bone, Blender will update the interface and you will see that the new bone's root moves to the tip of the parent bone.

Fill between roots.

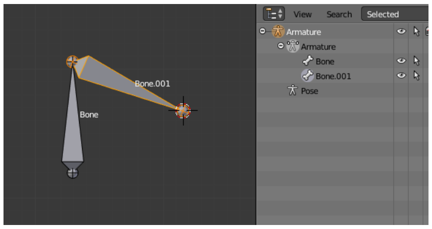

View
Search
Selected
Armature
Armature
Bone
Bone.002
Bone.001
Pose
Bone
Bone.001
Bone.002

Before UI update (3D cursor on the left).

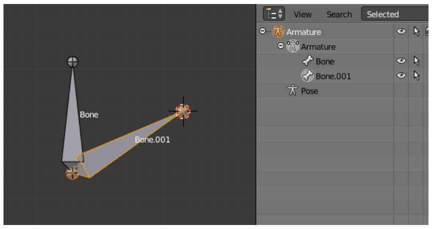
After UI update, correct visualization.

Clicking `F` with only one bone joint selected will create a bone from the selected joint to the 3D cursor position, and it will not parent it to any bone in the armature.

Fill with only one bone joint selected.


View
Search
Selected
Armature
Armature
Bone
Bone.001
Pose
Bone.001
Bone

Fill with only one tip selected.


View
Search
Selected
Armature
Armature
Bone
Bone.001
Pose
Bone
Bone.001

Fill with only one root selected.

You will get an error when:

- Trying to fill two joints of the same bone.
- Trying to fill more than two bone joints.

Split

Reference

Mode::	Edit Mode
Menu::	*Armature ‣ Split*
Shortcut::	Y

Disconnects the selection and clears the parent at the start and end. ToDo <2.8 add.

Separate Bones

Reference

Mode::	Edit Mode
Menu::	*Armature ‣ Separate Bones*
Shortcut::	`P`

You can, as with meshes, separate the selected bones in a new armature object *Armature ‣ Separate*, `Ctrl-Alt-P` and of course, in *Object Mode*, you can join all selected armatures in one *Object ‣ Join Objects*, `Ctrl-J`.

Subdivide

Reference

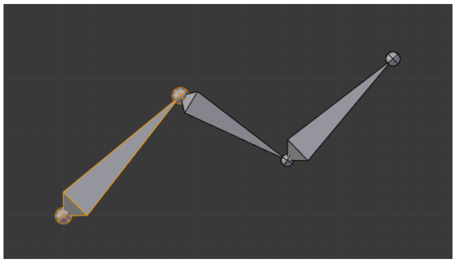
Mode:: Edit Mode
Menu:: *Armature* ‣ *Subdivide*

You can subdivide bones, to get two or more bones where there was just one bone. The tool will subdivide all selected bones, preserving the existing relationships: the bones created from a subdivision always form a connected chain of bones.

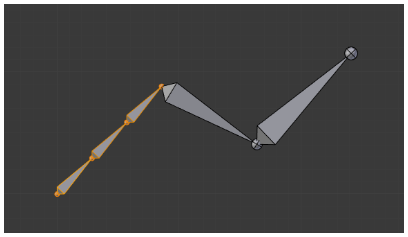
To create an arbitrary number of bones from each selected bone in the Subdivide Multi Adjust Last Operation panel.

Number of Cuts
  Specifies the number of cuts. As in mesh editing, if you set n cuts, you will get $n + 1$ bones for each selected bone.

Subdivision example.



An armature with one selected bone, just before multi-subdivision.



The selected bone has been “cut” two times, giving three sub-bones.

Switch Direction

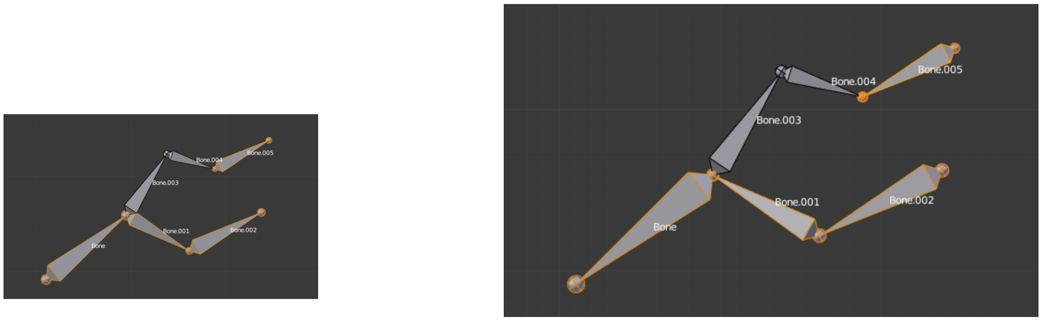
Reference

Mode:: Edit Mode
Menu:: *Armature ‣ Switch Direction*
Shortcut:: `Alt-F`

This tool allows you to switch the direction of the selected bones (i.e. their root will become their tip, and vice versa).

Switching the direction of a bone will generally break the chain(s) it belongs to. However, if you switch a whole (part of a) chain, the switched bones will still be parented/connected, but in "reversed order". See the Fig. Switching example..

Switching example.



An armature with one selected bone, and one selected chain of three bones, just before switching.

The selected bones have been switched. Bone.005 is no more connected nor parented to anything. The chain of switched bones still exists, but reversed (now Bone.002 is its root, and Bone is its tip). Bone.003 is now a free bone.

Symmetrize

Reference

Mode::	Edit Mode
Menu::	*Armature* ▸ *Symmetrize*

This operator will mirror the selected bones along the X axis based on Blender's bone naming convention for symmetrical armatures, either from left to right or right to left, depending on the selection.

Note

If the side of the bone cannot be determined, it will be ignored.

Bones with the opposite names that don't yet exist will be created, and already existing ones will be overwritten. If matching bones are selected on both sides, mirroring will happen from right to left.

Symmetrized bone and constraint properties will have the necessary changes to mirror their behaviors. When symmetrizing bones with Action Constraints, the necessary keyframes will be added to the target Action to result in symmetrical movement when the Action is activated.

Note

Note that bone or constraint drivers will not be created or affected in any way.

Naming

Reference

Mode:: All Modes
Panel:: *Properties ‣ Bone Properties*

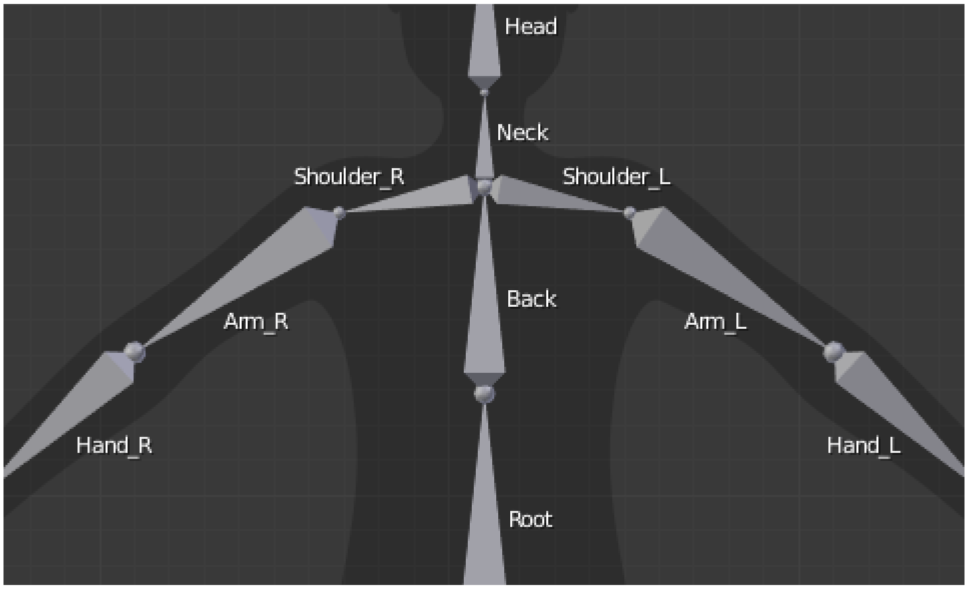
You can rename your bones, either using the *Name* field in the *Bones Properties*. It is also possible to rename by double-clicking bones in the Outliner.

Blender also provides you some tools that take advantage of bones named in a left/right symmetry fashion, and others that automatically name the bones of an armature.

Naming Conventions

Naming conventions in Blender are not only useful for you in finding the right bone, but also to tell Blender when any two of them are counterparts.

In case your armature can be mirrored in half (i.e. it is bilaterally symmetrical), it is worthwhile to stick to a left/right naming convention. This will enable you to use some tools that will probably save you time and effort (like the *X-Axis Mirror* editing tool).



An example of left/right bone naming in a simple rig.

1. First you should give your bones meaningful base-names, like “leg”, “arm”, “finger”, “back”, “foot”, etc.
2. If you have a bone that has a copy on the other side (a pair), like an arm, give it one of the following separators:

   - Left/right separators can be either the second position “L_calfbone” or last-but-one “calfbone.R”.
   - If there is a lower or upper case “L”, “R”, “left” or “right”, Blender handles the counterpart correctly. See below for a list of valid separators. Pick one and stick to it as close as possible when rigging; it will pay off.

Examples of valid separators:

- (nothing): handLeft –> handRight
- “_” (underscore): hand_L –> hand_R
- “.” (dot): hand.l –> hand.r
- “-” (dash): hand-l –> hand-r
- “ “ (space): hand LEFT –> hand RIGHT

Note

Note that all examples above are also valid with the left/right part placed before the name. You can only use the short “L”/ “R” code if you use a separator (e.g. “handL”/ “handR” will not work!).

3. Before Blender handles an armature for mirroring or flipping, it first removes the number extension, e.g. “.001”.
4. You can copy a bone named “bla.L” and flip it over using Flip Names. Blender will name the copy “bla.L.001” and flipping the name will give you “bla.R”.

Auto-Name

Reference

Mode::	Edit Mode
Menu::	*Armature* ▸ *Names* ▸ *Auto-Name Left/Right, Front/Back, Top/Bottom*

The three *AutoName* entries of the *Armature* ▸ *Names* menu allow you to automatically add a suffix to all selected bones, based on the position of their root relative to the armature’s origin and its local coordinates:

AutoName Left/Right

Will add the “.L” suffix to all bones with a *positive* X coordinate root, and the “.R” suffix to all bones with a *negative* X coordinate root. If the root is exactly at 0.0 on the X axis, the X coordinate of the tip is used. If both joints are at 0.0 on the X axis, the bone will just get a period suffix, with no “L”/ “R” (as Blender cannot decide whether it is a left or right bone…).

AutoName Front/Back

Will add the “.Bk” suffix to all bones with a *positive* Y coordinate root, and the “.Fr” suffix to all bones with a *negative* Y coordinate root. The same as with *AutoName Left-Right* goes for 0.0 Y coordinate bones…

AutoName Top/Bottom

Will add the “.Top” suffix to all bones with a *positive* Z coordinate root, and the “.Bot” suffix to all bones with a *negative* Z coordinate root. The same as with *AutoName Left-Right* goes for 0.0 Z coordinate bones…

Flip Names

Reference

Mode::	Edit Mode
Menu::	*Armature* ▸ *Names* ▸ *Flip Names*

You can flip left/right markers (see above) in selected bone names. This can be useful if you have constructed half of a symmetrical rig (marked for a left or right side) and duplicated and mirrored it, and want to update the names for the new side. Blender will swap text in bone names according to the above naming conventions, and remove number extensions if possible.

Parenting

Reference

Mode:: Edit Mode
Menu:: *Armature ‣ Parent*
Panel:: *Properties ‣ Bones ‣ Relations*
Shortcut:: `Ctrl-P`, `Alt-P`

You can edit the relationships between bones (and hence create/modify the chains of bones) both from the 3D Viewport and the Properties. Whatever method you prefer, it's always a matter of deciding, for each bone, if it has to be parented to another one, and if so, if it should be connected to it.

To parent and/or connect bones, you can:

In the *3D Viewport*, select the bone and *then* its future parent, and press `Ctrl-P` (or *Armature ‣ Parent ‣ Make Parent…*). In the small *Make Parent* menu that pops up, choose *Connected* if you want the child to be connected to its parent, else click on *Keep Offset*. If you have selected more than two bones, they will all be parented to the last selected one. If you only select one already-parented bone, or all selected bones are already parented to the last selected one, your only choice is to connect them, if not already done. If you select only one non-parented bone, you will get the *Need selected bone(s)* error message…

Note

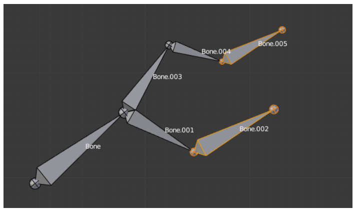
With this method, the newly-children bones will not be scaled nor rotated – they will just be moved if you choose to connect them to their parent's tip.

In the *Properties*, Bones tab, for each selected bone, you can select its parent in the *Parent* data ID to the upper right corner of its Relations panel. If you want them to be connected, just enable the checkbox to the right of the list.

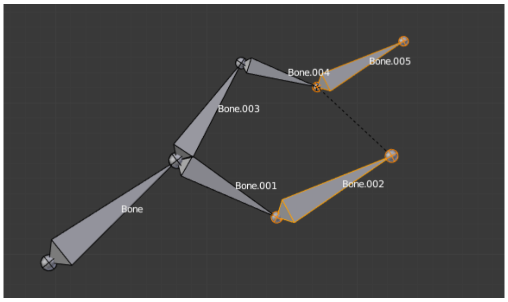
Note

With this method, the tip of the child bone will never be moved – so if *Connected* is enabled, the child bone will be completely transformed by the operation.

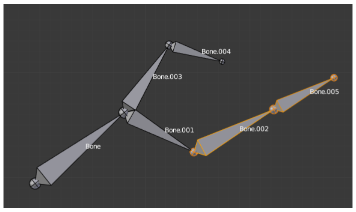
Parenting example.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

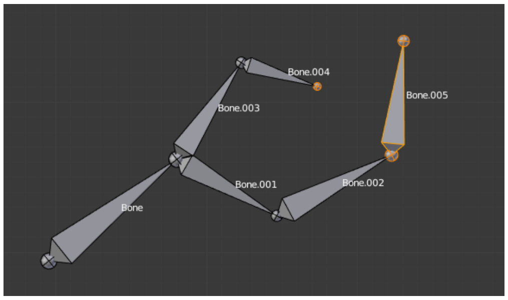
The starting armature, with Bone.005 parented and connected to Bone.004.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

Bone.005 re-parented to Bone.002, but not connected to it (same result, using either `Ctrl-P 2` in 3D Viewport, or the Bones tab settings).


Bone.004
Bone.003
Bone.005
Bone.001
Bone.002
Bone

Bone.005 parented and connected to Bone.002, using


Bone.004
Bone.003
Bone.005
Bone.001
Bone.002
Bone

Bone.005 parented and connected to Bone.002.

`Ctrl-P 1` in 3D Viewport.	Using the Parent data ID of Bone.005 Relations panel.

To disconnect and/or free bones, you can:

- In a 3D Viewport, select the desired bones, and press `Alt-P` (or *Armature* ▸ *Parent* ▸ *Clear Parent…*). In the small *Clear Parent* menu that pops up, choose *Clear Parent* to completely free all selected bones, or *Disconnect Bone* if you just want to break their connections.
- In the Properties, *Bones* tab, for each selected bone, you can select no parent in the *Parent* data ID of its Relations panel, to free it completely. If you just want to disconnect it from its parent, disable the *Connected* checkbox.

Note that relationships with non-selected children are never modified.

Bone Collections

Reference

Mode::	Edit Mode, Pose Mode
Menu::	*Armature* ▸ *Bone Collections*, *Pose* ▸ *Bone Collections*

Manages the Bone Collections the bone is assigned to.

Move to Collection `M`

Move bones to a collection.

Assign to Collection `Shift-M`

Assign all selected bones to a collection, or unassign them, depending on whether the active bone is already assigned or not.

Show All `Ctrl-AccentGrave`

Unhides any hidden bone collections.

Assign to new Collection

Assigns the selected bones to a new collection named “New Collection”. This collection can be renamed in the Bone Collections panel of the Armature properties.

Properties

Reference

Mode::	Edit Mode
Menu::	*Armature* ▸ *Bone Settings* ▸ …
Shortcut::	`Shift-W`, `Shift-Ctrl-W`, `Alt-W`

Most bones' properties (except the transform ones) are regrouped in each bone's panels, in the *Bones* tab in *Edit Mode*. Let us detail them.

Note that some of them are also available in the 3D Viewport, through the three pop-up menus within the same entry:

- *Toggle Setting*: `Shift-W` or *Armature* ▸ *Bone Settings* ▸ *Toggle a Setting*
- *Enable Setting*: `Shift-Ctrl-W` or *Armature* ▸ *Bone Settings* ▸ *Enable a Setting*
- *Disable Setting*: `Alt-W` or *Armature* ▸ *Bone Settings* ▸ *Disable a Setting*

Display Wire

Always display the bone as wireframe.

Deform

(also `Shift-W` ▸ *(Deform, …)*).

Multiply Vertex Group by Envelope

(also `Shift-W` ▸ *(Multiply Vertex Group by Envelope, …)*).

These settings control how the bone influences its geometry, along with the bones' joints radius. This will be detailed in the skinning part.

Inherit Rotation

The bone automatically rotates together with its parent in *Pose Mode*. For more details, see the relations page.

Lock

(also `Shift-W` ▸ *(Locked, …)*) This will prevent all editing of the bone in *Edit Mode*; see bone locking.

Delete

Bones

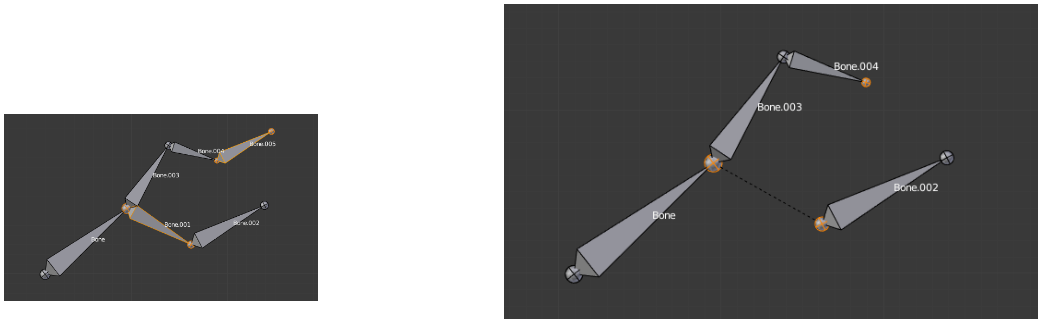
Reference

Mode::	Edit Mode
Menu::	*Armature ▸ Delete ▸ Bones*
Shortcut::	`X`

This tool delete selected bones, selected *joints* are ignored.

If you delete a bone in a chain, its child(ren) will be automatically re-parented to its own parent, but **not** connected, to avoid deforming the whole armature.

Deletion example.


Bone.005
Bone.004
Bone.003
Bone.001
Bone.002
Bone
Bone.004
Bone.003
Bone.002
Bone

An armature with two selected bones, just before deletion.

The two bones have been deleted. Note that Bone.002, previously connected to the deleted Bone.001, is now parented but not connected to Bone.

Dissolve

Reference

Mode::	Edit Mode
Menu::	*Armature* ‣ *Delete* ‣ *Dissolve*
Shortcut::	`Ctrl-X`

Todo 2.76.

Properties

Reference

Mode:: Object Mode, Edit Mode and Pose Mode
Panel:: *Properties* ‣ *Bone*

When bones are selected (hence in *Edit Mode* and *Pose Mode*), their properties are shown in the *Bone* tab of the Properties. This shows different panels used to control features of each selected bone; the panels change depending on which mode you are working in.

- Transform
- Bendy Bones
  - Technical Details
  - Display
  - Rest Pose
  - Example
  - Options
- Relations
  - Parenting
  - Bone Collections
- Inverse Kinematics
- Deform
  - Envelope
- Viewport Display
  - General
  - Bone Colors
  - Custom Shape
- Custom Properties

Transform

Reference

Mode::	Edit Mode and Pose Mode
Panel::	*Bone ‣ Transform*

When in *Edit Mode* you can use this panel to control position and roll of individual bones. Whereas in *Pose Mode* you can only set location for the main bone, and you can now set rotation and scale.

In addition, in *Pose Mode* it is possible to restrict changes in position, rotation and scale by axis on each bone in the armature.

Head X, Y, Z
: Location of head end of the bone.

Tail X, Y, Z
: Location of tail end of the bone.

Roll
: Bone rotation around head-tail axis.

Length
: The distance from the bone's head to it's tail. Changing the length moves the tail end.

Lock
: Bone is not able to be transformed when in Edit Mode.

Bendy Bones

Reference

Mode:: All Modes
Panel:: *Bone ‣ Bendy Bones*

Bendy Bones (B-Bones) are an easy way to replace long chains of many small rigid bones. A common use case for curved bones is to model spine columns or facial bones.

Technical Details

Blender treats the bone as a section of a Bézier curve passing through the bones' joints. Each of the *Segments* will bend and roll to follow this invisible curve representing a tessellated point of the Bézier curve. The control points at each end of the curve are the endpoints of the bone. The shape of the B-Bones can be controlled using a series of properties or indirectly through the neighboring bones (i.e. first child and parent). The properties construct handles on either end of the bone to control the curvature.

When using the B-bone as a constraint target Data ID offers an option to follow the curvature.

Note

However, if the bone is used as a target rather than to deform geometry, only *Armature* and *Copy Transforms* constraints will use the full transformation including roll and scale.

Display

You can see these segments only if bones are visualized as *B-bones*.

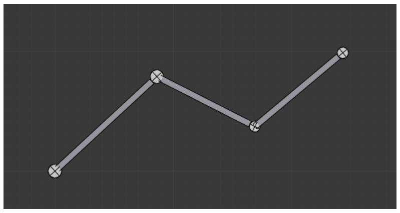
When not visualized as *B-Bone*s, bones are always shown as rigid sticks, even though the bone segments are still present and effective. This means that even in e.g. *Octahedron* visualization, if some bones in a chain have several segments, they will nonetheless smoothly deform their geometry.

Rest Pose

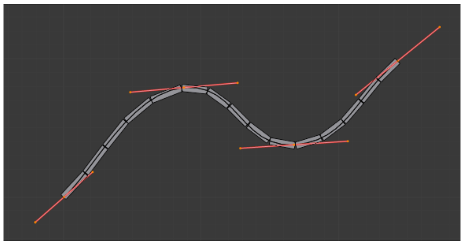
The initial shape of a B-Bone can be defined in Edit Mode as a rest pose of that bone. This is useful for curved facial features like curved eyebrows or mouths.

B-Bones have two sets of the Bendy Bone properties – one for Edit Mode (i.e. the Rest Pose/Base Rig) and another for Pose Mode – adding or multiplying together their values to get the final transforms.

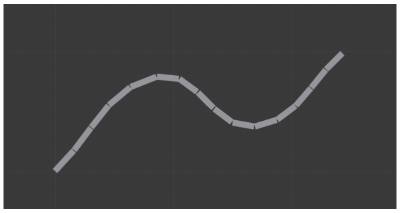
Example



Bones with just one segment in Edit Mode.



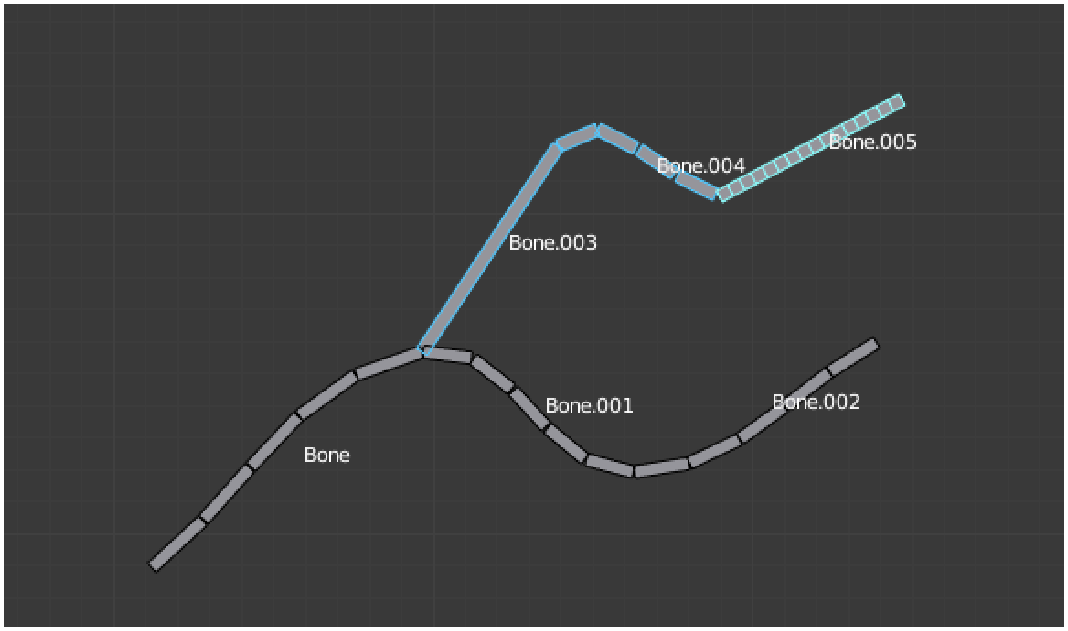
The Bézier curve superposed to the chain, with its handles placed at bones' joints.



The same armature in Object Mode.

In Fig. Bones with just one segment in Edit Mode. we connected three bones, each one made of five segments.

Look at Fig. The same armature in Object Mode., we can see how the bones' segments smoothly "blend" into each other, even for roll.



An armature in Pose Mode, B-Bone visualization: Bone.003 has one segment, Bone.004 has four, and Bone.005 has sixteen.

Options

Segments

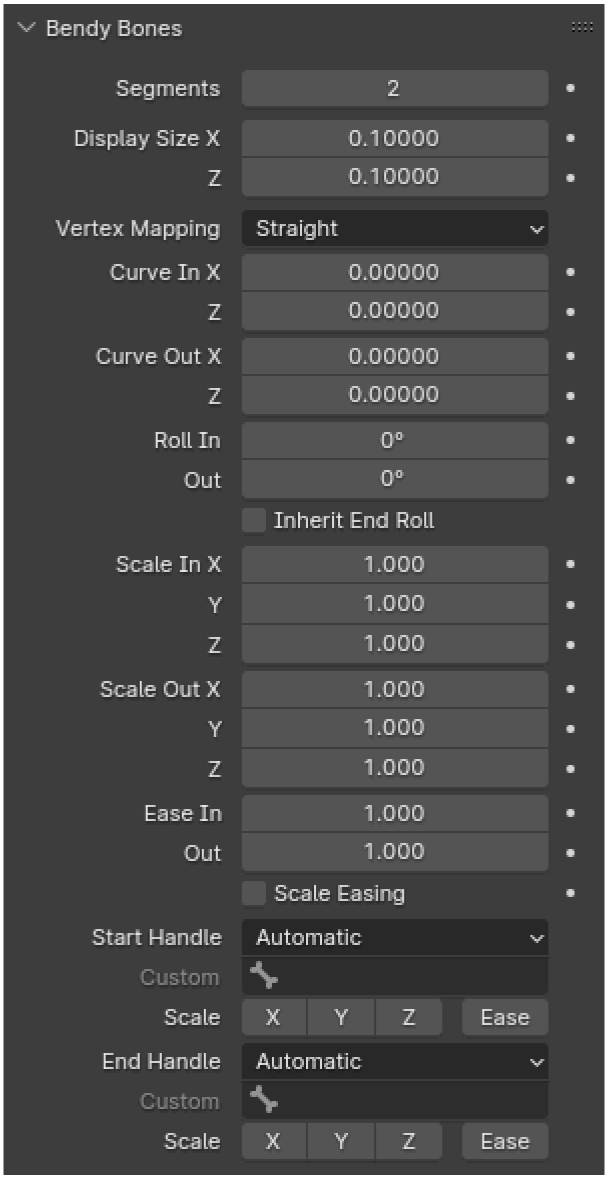
The number of segments, which the given bone is subdivided into. Segments are small, rigid linked child bones that interpolate between the root and the tip. The higher this setting, the smoother "bends" the bone, but the heavier the pose calculations.

Display Size X, Z

Controls the visible thickness of the bone segments when the armature is rendered in the *B-Bones* mode.

Vertex Mapping

Controls how vertices are weighted to the individual segments of a B-Bone for deformations:



Segments 2
Display Size X 0.10000
Z 0.10000
Vertex Mapping Straight
Curve In X 0.00000
Z 0.00000
Curve Out X 0.00000
Z 0.00000
Roll In 0°
Out 0°
☐ Inherit End Roll
Scale In X 1.000
Y 1.000
Z 1.000
Scale Out X 1.000
Y 1.000
Z 1.000
Ease In 1.000
Out 1.000
☐ Scale Easing
Start Handle Automatic
Custom
End Handle Automatic
Custom

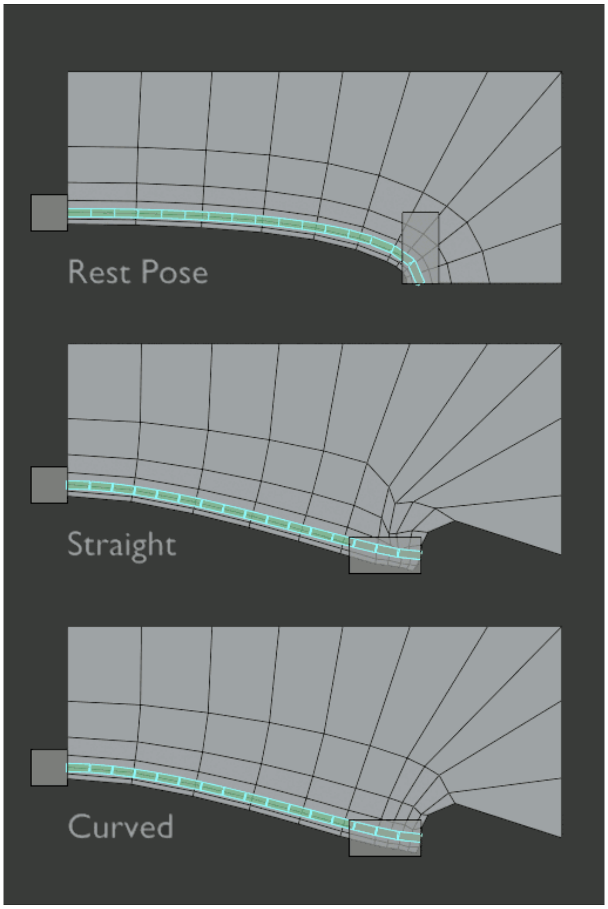
Bendy Bones panel.

Straight::

A fast mapping that works well for B-Bones with a straight or gently curved rest pose.

Curved::

A slower mapping that improves deformations for B-Bones with a strongly curved rest pose. This should be used selectively when needed.



Straight vs Curved vertex mapping on a B-Bone with a strongly curved rest pose.

Curve In/Out X, Y, Z
: Applies offsets to the curve handle positions on the plane perpendicular to the bone's primary (Y) axis. As a result, the handle moves per axis (XZ) further from its original location, causing the curve to bend.

Roll In, Out
: The roll value (or twisting around the main Y axis of the bone) is interpolated per segment, between the start and end roll values. It is applied as a rotational offset on top of the rotation defined by the handle bones.

Inherit End Roll
: If enabled, the *Roll Out* value of the *Start Handle* bone (connected parent by default) will be implicitly added to the *Roll In* setting of the current bone.

Scale In/Out X, Y, Z

Scaling factors that adjust the thickness of each segment for the X and Z axes, or introduce non-uniform spacing along the Y axis. Similar to *Roll* it is interpolated per segment.

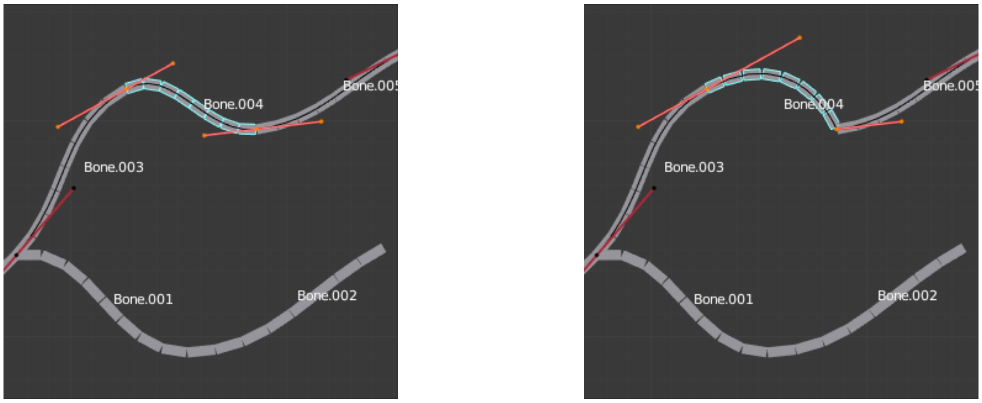
Since all segments are still uniformly scaled in the Y direction to fit the actual length of the curve, only the ratio between *Scale In Y* and *Scale Out Y* actually matters.

Ease In, Out

The *Ease In/Out* number fields, change the "length" of the "auto" Bézier handle to control the "root handle" and "tip handle" of the bone, respectively. These values are proportional to the default length, which of course automatically varies depending on bone length, angle with the reference handle, and so on.

Although easing is a scale-like value, the Edit Mode and Pose Mode versions of the values are added, so they get corresponding start values of 1 and 0 by default.

Ease In/Out settings example, with a materialized Bézier curve.



Bone.004 with default In and Out (1.0).

Bone.004 with In at 2.0, and Out at 0.0.

Scale Easing

If enabled, the final easing values are implicitly multiplied by the corresponding *Scale Y* values.

Custom Handles

B-Bones can use custom bones as their reference bone handles, instead of only using the connected parent/child bones.

Start/End Handle

Specifies the type of the handle from the following choices:

Automatic::	The connected parent (or first connected child) of the bone is chosen as the handle. Calculations are done according to the *Absolute* handle type below.
Absolute::	The Bézier handle is controlled by the **position** of the head (tail) of the handle bone relative to the head (tail) of the current bone. Note that for this to work, there must be a nonzero distance between these bones. If the handle is also a B-Bone, additional processing is applied to further smooth the transition, assuming that the bones in effect form a chain.
Relative::	The Bézier handle is controlled by the **offset** of the head (tail) of the handle bone from its rest pose. The use of this type is not recommended due to numerical stability issues near zero offset.
Tangent::	The Bézier handle is controlled by the **orientation** of the handle bone, independent of its location.

Custom Handle

For types other than *Automatic*, a bone to use as handle has to be manually selected. Switching to a custom handle type without selecting a bone can be used to effectively disable the handle.

It is valid for two bones to refer to each other as handles – this correlation is applied in connected chains with *Automatic* handles.

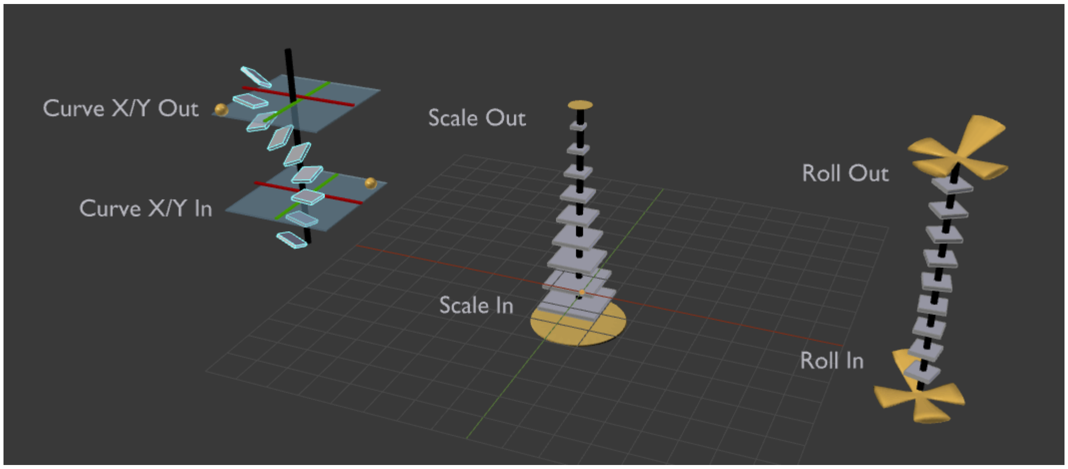
Scale X/Y/Z/Ease

If enabled, the final Scale and/or Ease values are multiplied by the corresponding local scale channels of the handle bone. This step is applied independently of *Scale Easing* and doesn't interact with it, i.e. enabling *Y* and *Scale Easing* doesn't replace the *Ease* toggle. These toggles are a more efficient replacement for up to eight trivial drivers passing segment scale data from the handle bones into the B-Bone option properties.

Tip

Keying Set

The "BBone Shape" Keying Set includes all Bendy Bones properties.



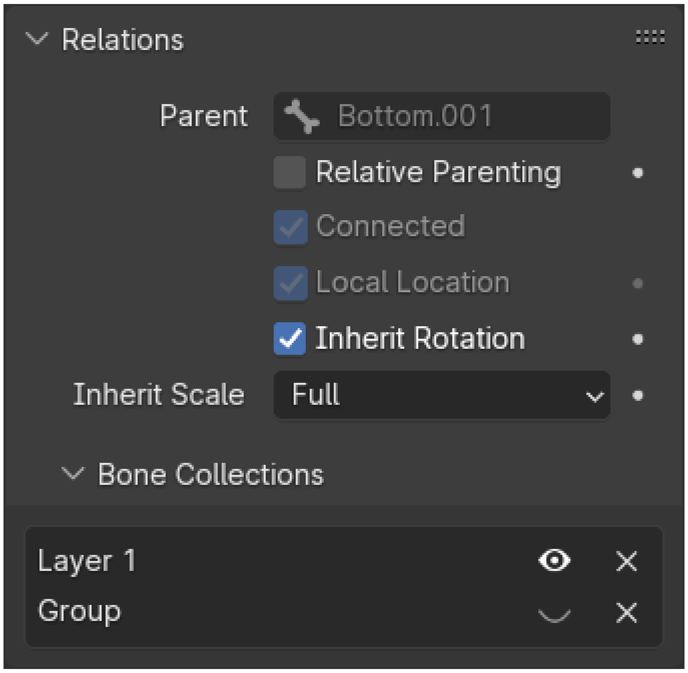
Visualization of the Bendy Bones properties.

From Left: 1) Curve X/Y offsets, 2) Scale In/Out, 3) Roll In/Out

Relations

Reference

Mode:: All Modes
Panel:: *Bone ▸ Relations*



Relations

Parent: Bottom.001
☐ Relative Parenting
☑ Connected
☑ Local Location
☑ Inherit Rotation
Inherit Scale: Full

Bone Collections

Layer 1
Group

Bone Relations panel.

In this panel you can manage the relationship of this bone with its parent bone. It also shows the bone collections the bone is assigned to.

Parenting

Parent
A Data ID to select the bone to set as a parent.

Relative Parenting *Pose Mode Only*

Changes how transformation of the bone is applied to its child Objects.

Connected
The *Connected* checkbox set the head of the bone to be connected with its parent root.

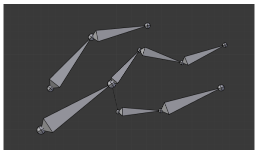
Transformations

Bones relationships have effects on transformations behavior.

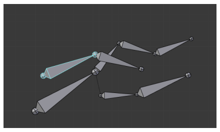
By default, children bones inherit:

- Their parent position, with their own offset of course.
- Their parent rotation (i.e. they keep a constant rotation relatively to their parent).
- Their parent scale, here again with their own offset.

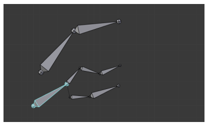
Examples of transforming parented/connected bones.



The armature in its rest position.



Rotation of a root bone.



Scaling of a root bone.

Exactly like standard children objects. You can modify this behavior on a per-bone basis, using the Relations panel in the *Bones* tab:

Local Location
When disabled, the location transform property is evaluated in the parent bone's local space, rather than using the bone's own *rest pose* local space orientation.

Inherit Rotation

When disabled, this will “break” the rotation relationship to the bone’s parent. This means that the child will keep its rotation in the armature object space when its parent is rotated.

Inherit Scale

Specifies which effects of parent scaling the bone inherits:

These inheriting behaviors propagate along the bones’ hierarchy. So when you scale down a bone, all its descendants are by default scaled down accordingly. However, if you disable one bone’s *Inherit Scale* or *Inherit Rotation* property in this “family”, this will break the scaling propagation, i.e. this bone *and all its descendants* will no longer be affected when you scale one of its ancestors.

Full::	The bone inherits all effects of parent scaling and shear.
Fix Shear::	Full parent effects are applied to the rest state of the child, after which any shear is removed in a way that preserves the bone direction, length and volume, and minimally affects roll on average. The result is combined with the local transformation of the child. If the inherited scale is non-uniform, this does not prevent shear from reappearing due to local rotation of the child bone, or of its children.
Aligned::	Parent scaling is inherited as if the child was oriented the same as the parent, always applying parent X scale over child X scale, and so on.
Average::	Inherits a uniform scaling factor that is the total change in the volume of the parent.
None::	Ignores all scaling and shear of the parent.
None (Legacy)::	Ignores all scaling, provided the parent is not sheared. If it is, there are no guarantees. This choice replicates the behavior of the old Inherit Scale checkbox, and may be removed in a

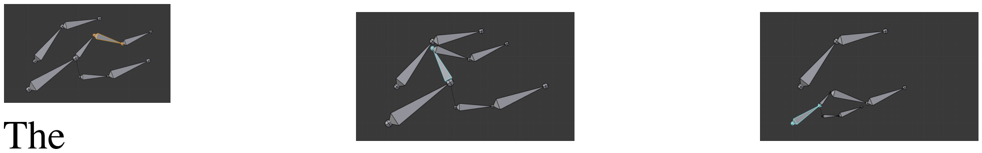
future release.

Tip

The various *Inherit Scale* options are provided as tools in avoiding shear that is caused by non-uniform scaling combined with parenting and rotation. There is no obvious best way to achieve that, so different options are useful for different situations.

- **None** – Useful for gaining full control over the scaling of the child in order to e.g. manually overwrite it with constraints.
- **Average** – Useful to block squash and stretch propagation between sub-rigs, while allowing uniform changes in the size and volume to pass through.
- **Aligned** – Can be used within bone chains, e.g. tentacles, in order to propagate lengthwise scaling as lengthwise, and sideways as sideways, no matter how the tentacle bends. Similar to using *None* with Copy Scale from parent.
- **Fix Shear** – May be useful at the base of an appendage in order to reallocate squash and stretch between axes based on the difference in rest pose orientations of the parent and child. It behaves closest to *Full* while suppressing shear.

Examples of transforming parented/connected bones with Inherit Rotation disabled.



The yellow outlined Inherit Rotation disabled bone in the armature.

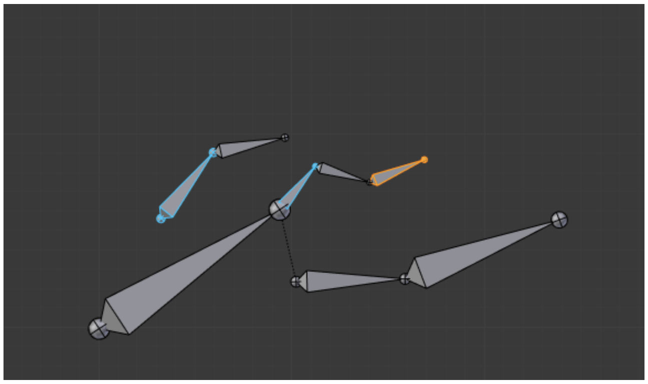
Rotation of a bone with an Inherit Rotation disabled bone among its descendants.

Scaling of a bone with an Inherit Rotation disabled bone among its descendants.

Connected bones have another specificity: they cannot be moved. Indeed, as their root must be at their parent's tip, if you do not move the parent, you cannot move the child's root, but only its tip, which leads to a child rotation. This is exactly what happens, when you press `G` with a connected bone selected, Blender automatically switches to rotation operation.

Bones relationships also have important consequences on how selections of multiple bones behave when transformed. There are many different situations which may not be included on this list, however, this should give a good idea of the problem:

- Non-related selected bones are transformed independently, as usual.
- When several bones of the same "family" are selected, *only* the "most parent" ones are really transformed – the descendants are just handled through the parent relationship process, as if they were not selected (see Fig. Scaling bones, some of them related. the third tip bone, outlined in yellow, was only scaled down through the parent relationship, exactly as the unselected ones, even though it is selected and active. Otherwise, it should have been twice smaller!)



Scaling bones, some of them related.

- When connected and unconnected bones are selected, and you start a move operation, only the unconnected bones are affected.
- When a child connected hinge bone is in the selection, and the "most parent" selected one is connected, when you press `G`, nothing happens,

because Blender remains in move operation, which of course has no effect on a connected bone.

So, when posing a chain of bones, you should always edit its elements from the root bone to the tip bone. This process is known as Forward Kinematics (FK). We will see in a later page that Blender features another pose method, called Inverse Kinematics (IK), which allows you to pose a whole chain just by moving its tip.

Note

This feature is somewhat extended/completed by the pose library.

Bone Collections

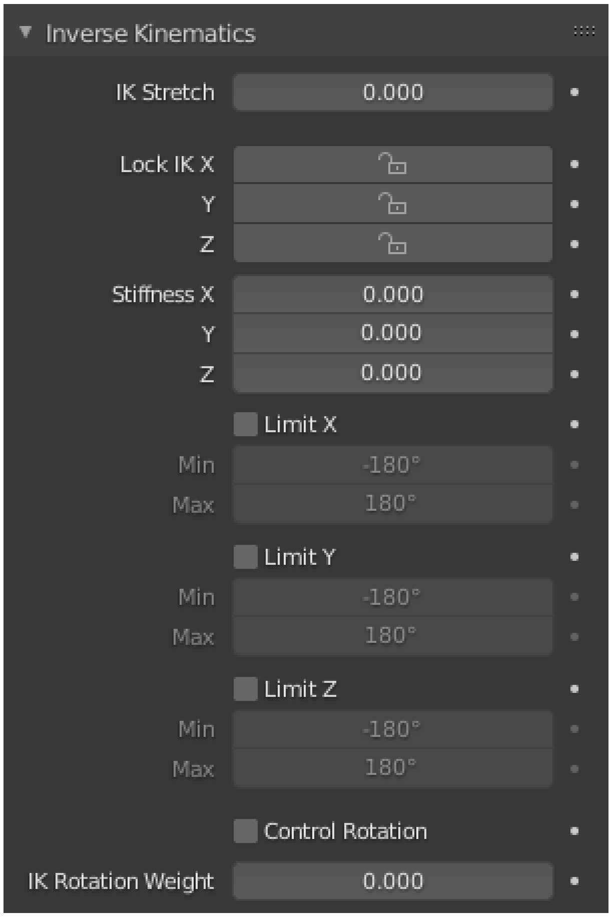
This list shows the bone collections the bone is assigned to. Press the eye icon to show or hide the entire bone collection. Press the star icon to show only this bone collection, and others also marked as ‘solo’ Press the X icon to remove the bone from that particular collection.

To assign the bone to other bone collections, either use the `M` or `Shift-M` shortcuts (see Moving Bones Between Collections) or go to the Armature properties panel.

Inverse Kinematics

Reference

Mode:: Pose Mode
Panel:: *Bone* ▸ *Inverse Kinematics*



IK Stretch 0.000
Lock IK X 🔓
Y 🔓
Z 🔓
Stiffness X 0.000
Y 0.000
Z 0.000
☐ Limit X
Min -180°
Max 180°
☐ Limit Y
Min -180°
Max 180°
☐ Limit Z
Min -180°
Max 180°
☐ Control Rotation
IK Rotation Weight 0.000

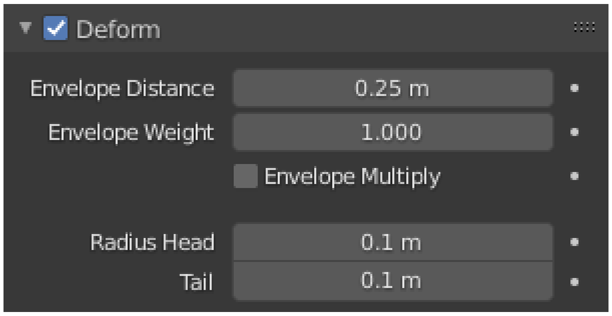
The Inverse Kinematics panel.

This panel controls the way a bone or set of bones behave when linked in an inverse kinematic chain.

Deform

Reference

Mode:: All Modes
Panel:: *Bone ‣ Deform*



☑ Deform

Envelope Distance 0.25 m

Envelope Weight 1.000

☐ Envelope Multiply

Radius Head 0.1 m

Tail 0.1 m

The Deform panel.

In this panel you can set deformation options for each bone.

Toggling the checkbox in the panel header off, prevents the bone from deforming the geometry at all, overriding any weights that it might have been assigned before; it mutes its influence.

It also excludes the active bone in the automatic weight calculation when the mesh is parented to the armature using the *Armature Deform* tool with the *With Automatic Weights* option.

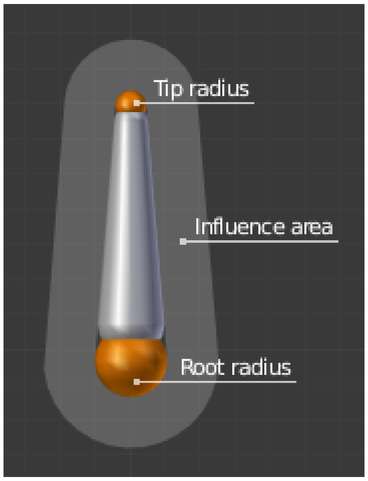
Envelope

Envelopes is the most general skinning method. It works with all available object types for skinning (meshes, lattices, curves, surfaces and texts). It is based on proximity between bones and their geometry, each bone having two different areas of influence, shown in the *Envelope* visualization:

- The inside area, materialized by the “solid” part of the bone, and controlled by both root and tip radius.
- The outside area, materialized by the lighter part around the bone, and controlled by the *Distance* setting.

See also

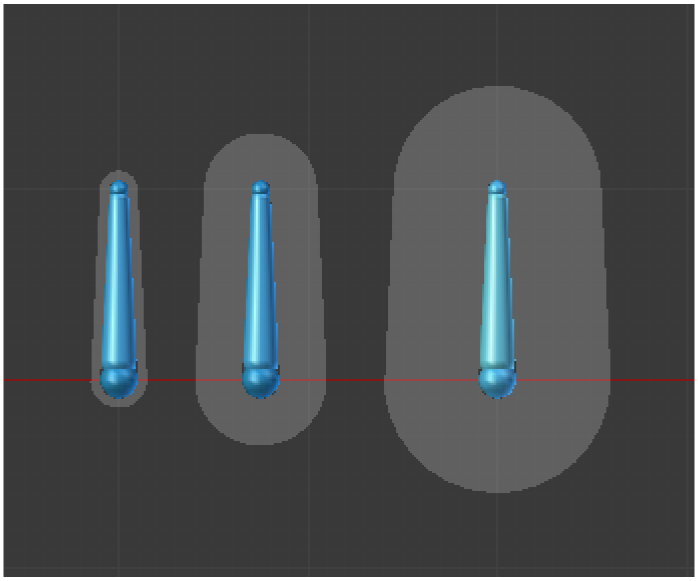
The editing pages for how to edit these properties.



Bone influence areas for envelopes method.

Envelope Distance

The Distance defines a volume which is the range within the bone has an influence on vertices of the deformed object. The geometry is less and less affected by the bone as it goes away by following a quadratic decay.



Single bone with various envelope sizes.

Envelope Weight

A bone property, that controls the global influence of the bone over the deformed object, when using the envelopes method.

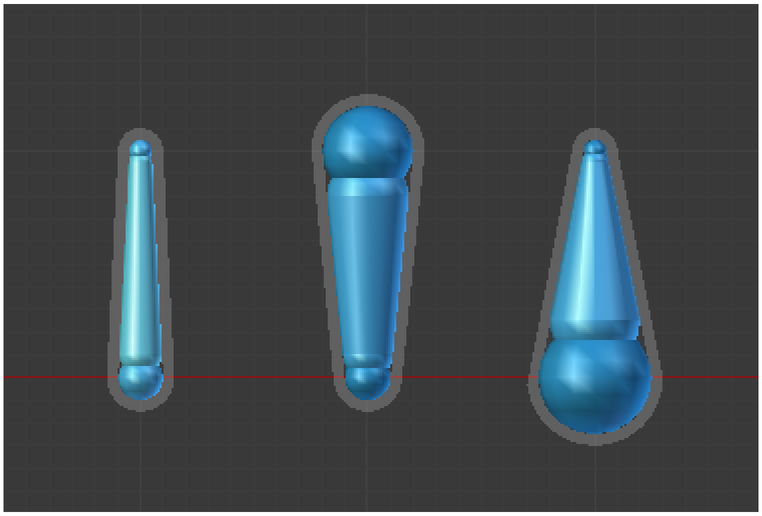
It is only useful for the parts of geometry that are "shared", influenced by more than one bone (generally, at the joints…) – a bone with a high weight will have more influence on the result than one with a low weight… Note that when set to 0.0, it has the same effect as disabling the *Deform* option.

Envelope Multiply

This option controls how the two deforming methods interact, when they are both enabled. By default, when they are both active, all vertices belonging to at least one vertex group are only deformed through the vertex groups method. The other "orphan" vertices being handled by the envelopes one. When you enable this option, the "deformation influence" that this bone would have on a vertex (based from its envelope settings) is multiplied with this vertex's weight in the corresponding vertex group. In other words, the vertex groups method is further "weighted" by the envelopes method.

Radius Head, Tail

Set the radius for the head and the tail of envelope bones. Inside this volume, the geometry if fully affected by the bone.



Three Armature Bones all using Envelope Weight.

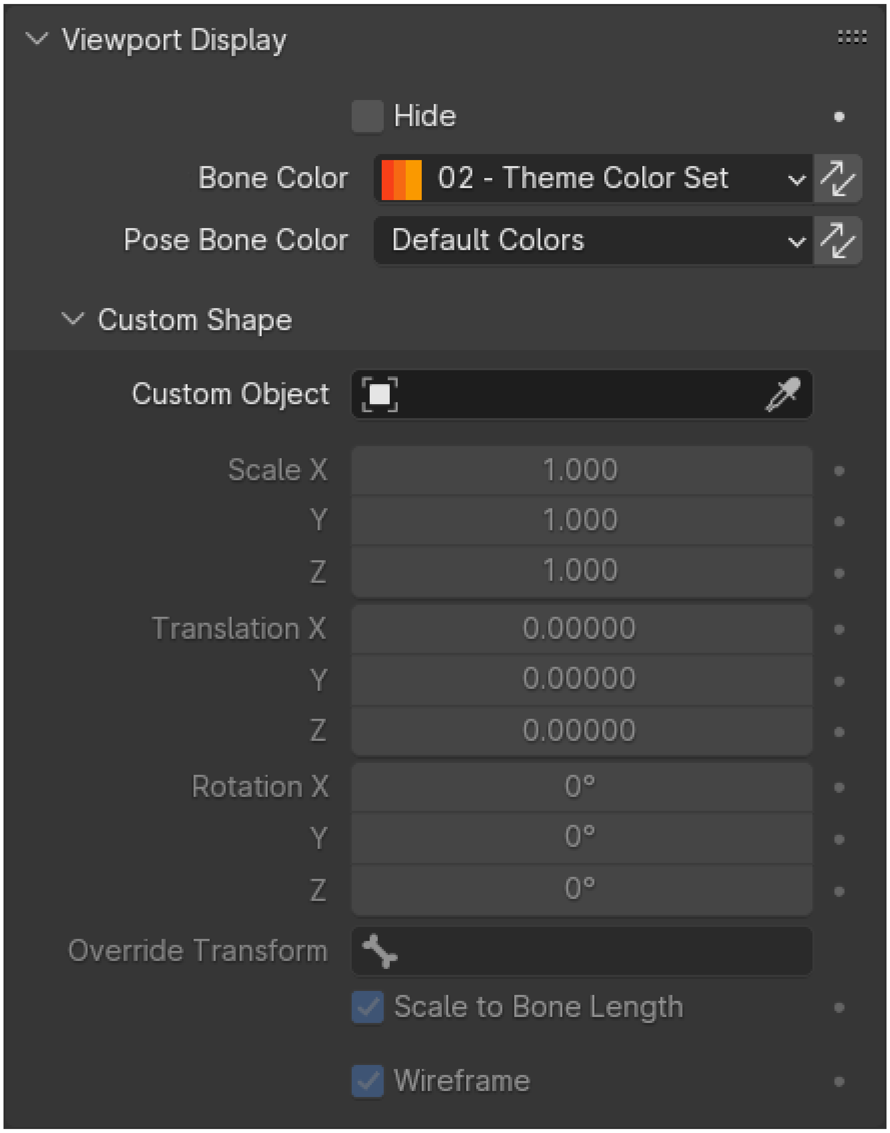
The 1st with a default radius value, the two others with differing Tail and Head radius values.

Viewport Display

Reference

Mode::	Object, Pose, and Edit Mode
Panel::	*Bone* ‣ *Viewport Display*

This panel lets you customize the look of your bones.



Viewport Display

☐ Hide

Bone Color: 02 - Theme Color Set

Pose Bone Color: Default Colors

Custom Shape

Custom Object

Scale X: 1.000
Y: 1.000
Z: 1.000

Translation X: 0.00000
Y: 0.00000
Z: 0.00000

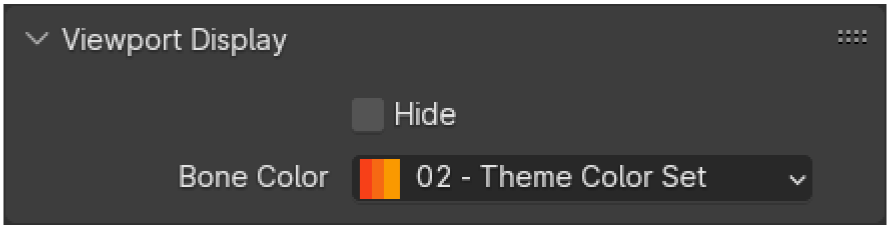
Rotation X: 0°
Y: 0°
Z: 0°

Override Transform

☑ Scale to Bone Length

☑ Wireframe

Viewport Display panel in Object/Pose mode.



☐ Hide

Bone Color: 02 - Theme Color Set

Viewport Display panel in Edit mode.

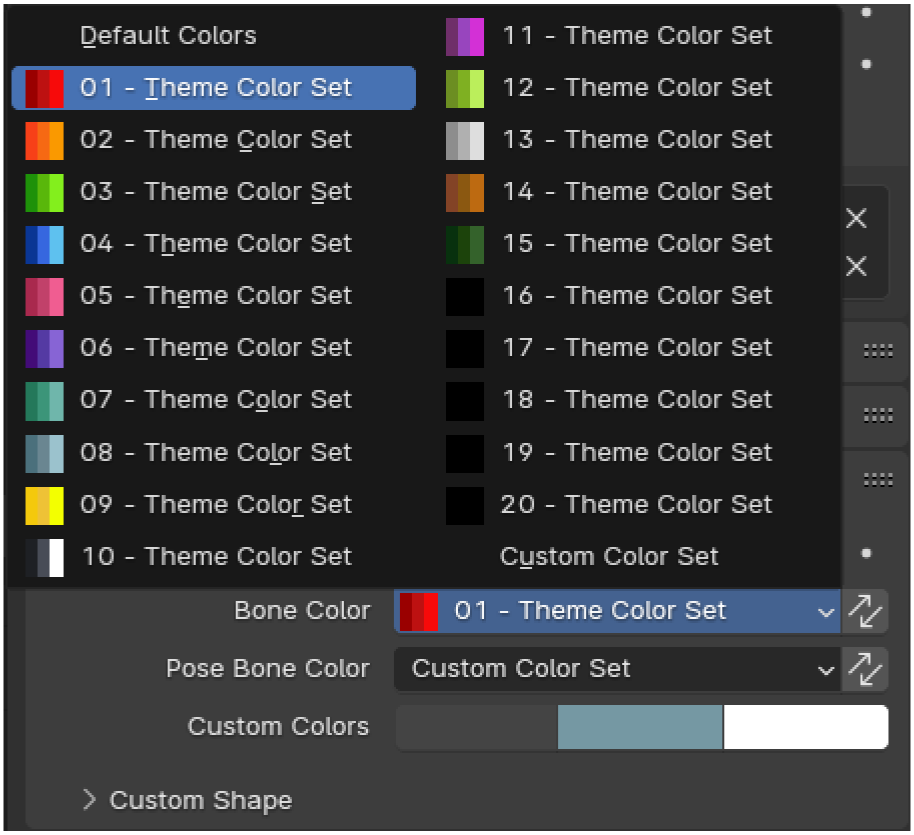
General

Hide

Hides the bone in the 3D Viewport. When this is unchecked, the bone’s visibility is determined by the visibility of its bone collections.

Bone Colors

Bones can be individually colored. You can either choose a color set from the predefined theme list or define a custom one.


Default Colors
01 - Theme Color Set
02 - Theme Color Set
03 - Theme Color Set
04 - Theme Color Set
05 - Theme Color Set
06 - Theme Color Set
07 - Theme Color Set
08 - Theme Color Set
09 - Theme Color Set
10 - Theme Color Set
11 - Theme Color Set
12 - Theme Color Set
13 - Theme Color Set
14 - Theme Color Set
15 - Theme Color Set
16 - Theme Color Set
17 - Theme Color Set
18 - Theme Color Set
19 - Theme Color Set
20 - Theme Color Set
Custom Color Set
Custom Colors
Custom Shape

Bone Color: 01 - Theme Color Set

Pose Bone Color: Custom Color Set

When selecting *Custom Color Set*, you need to define three colors: Regular (for when the bone is not selected), Selected, and Active.

You can temporarily disable all the color assignments by unchecking Bone Colors in the armature's Viewport Display panel.

Bone Color

The bone's primary color, affecting both Edit Mode and Pose Mode.

This color is stored on the armature data-block, so that if you have multiple armature objects that share this data-block, they will all use the same color.

Copy Bone Color to Selected

Copy the bone color of the Active bone to all selected bones.

Pose Bone Color *Pose Mode*

Lets you optionally override the above *Bone Color* in Pose Mode (by setting it to something else than *Default Colors*).

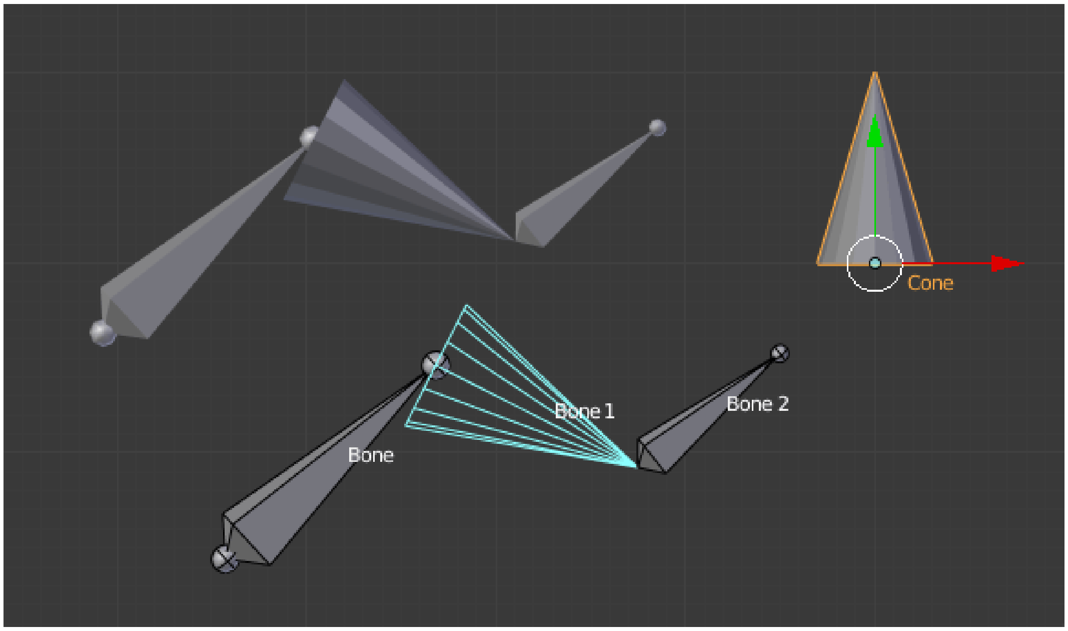
This color is stored on the Pose Bone, meaning it can be different in every armature object – even ones that reference the same data-block.

Copy Bone Color to Selected

Copy the bone color of the Active bone to all selected bones.

Custom Shape

Apart from custom colors, bones can also have custom shapes (in *Object Mode* and *Pose Mode*), using another object as a "template."



A bone referencing a cone as its Custom Shape.

You can temporarily disable these shapes by unchecking *Shapes* in the armature's Viewport Display panel.

Custom Object

Object that defines the custom shape of the selected bone.

Scale X, Y, Z

Additional scaling factor to apply to the custom shape.

Translation X, Y, Z

Additional translation to apply to the custom shape.

Rotation X, Y, Z

Additional rotation to apply to the custom shape.

Override Transform

Bone that defines the display transform of the custom shape.

Scale to Bone Length

Whether the custom shape should be scaled by a factor equal to the bone's length.

Wireframe

When enabled, the bone is displayed in wireframe mode regardless of the viewport's shading mode.

Wire Width

The line thickness of the wireframe for the custom shape.

Note

- Custom shapes will never be rendered. Like regular bones, they are only visible in the 3D Viewport.
- The transforms of the template object are ignored. Moving, rotating, or scaling it will have no effect on its appearance in the armature.
- The origin of each instanced shape object is at the root of the bone.
- The rotation of each shape object is such that its Y axis lies along the direction of the bone.
- For best results when Scale to Bone Length is enabled, make sure the template object is 1 unit in size along its Y axis. This will make it perfectly match the size of each bone.

Custom Properties

Reference

Mode::	Pose Mode
Panel::	*Bone* ▸ *Custom Properties*

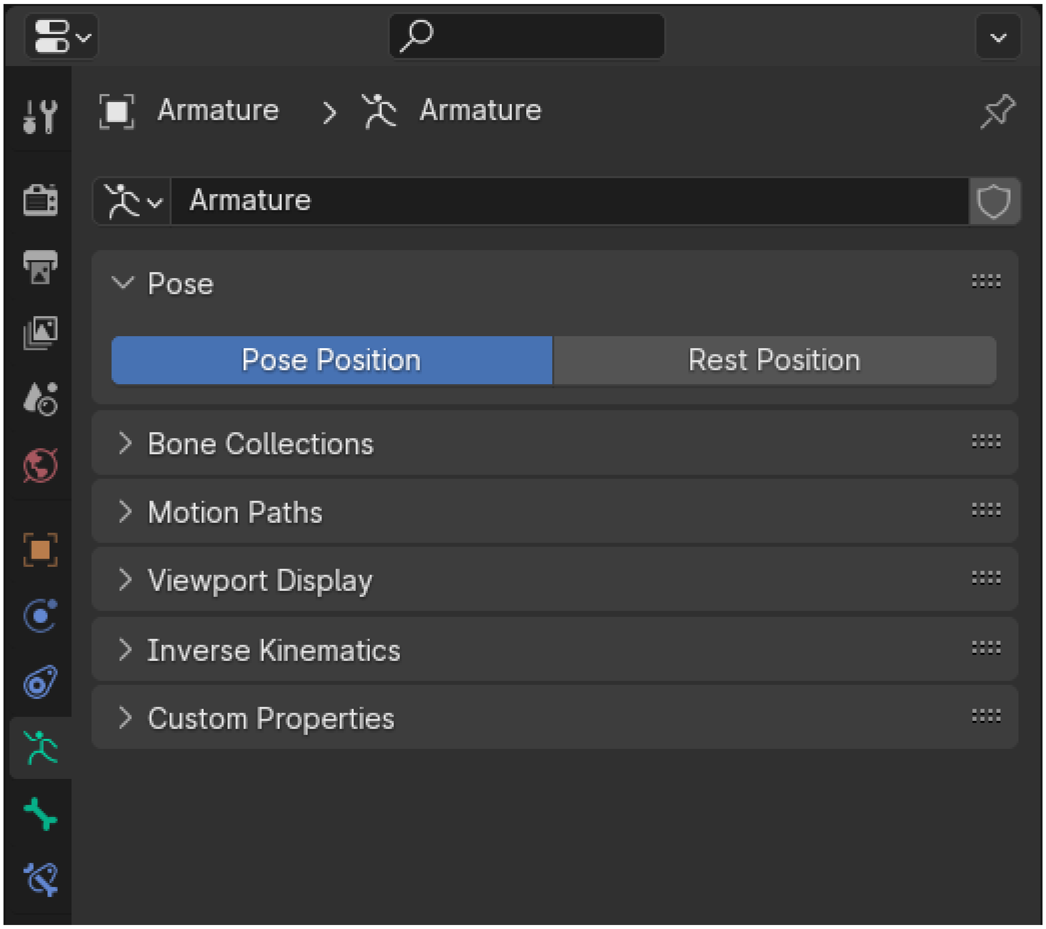
See the Custom Properties page for more information.

Properties

- Introduction
  - Pose
  - Bone Collections
  - Motion Paths
  - Inverse Kinematics
  - Custom Properties
- Bone Collections
  - Specials
  - Assign & Select
  - Moving Bones between Collections
  - Custom Properties
- Selection Sets
- Viewport Display

Introduction

The *Armature* tab in Properties contains various panels gathering the armature settings.


Armature
Armature
Armature
Pose
Pose Position
Rest Position
Bone Collections
Motion Paths
Viewport Display
Inverse Kinematics
Custom Properties

The Armature tab in the Properties.

Pose

Reference

Mode::	All Modes
Panel::	*Armature* ‣ *Pose*

Pose Position

A radio button to switch between Pose Position and Rest Position.

In *Edit Mode*, you always see armatures in their rest position, in *Object Mode* and *Pose Mode*, by default, you see them in *Pose Position* (i.e. as it was transformed in the *Pose Mode*). If you want to see it in the rest position in all modes, select *Rest Position*.

Bone Collections

See Bone Collections.

Motion Paths

Reference

Mode::	All Modes
Panel::	*Armature* ▸ *Motion Paths*

In the Motion Paths panel you can enable visualization of the motion path your skeleton leaves when animated.

Inverse Kinematics

Reference

Mode::	All Modes
Panel::	*Armature* ▸ *Inverse Kinematics*

Defines the type of IK solver used in your animation.

Custom Properties

Reference

Mode:: All Modes
Panel:: *Armature* ▸ *Custom Properties*

See the Custom Properties page for more information.

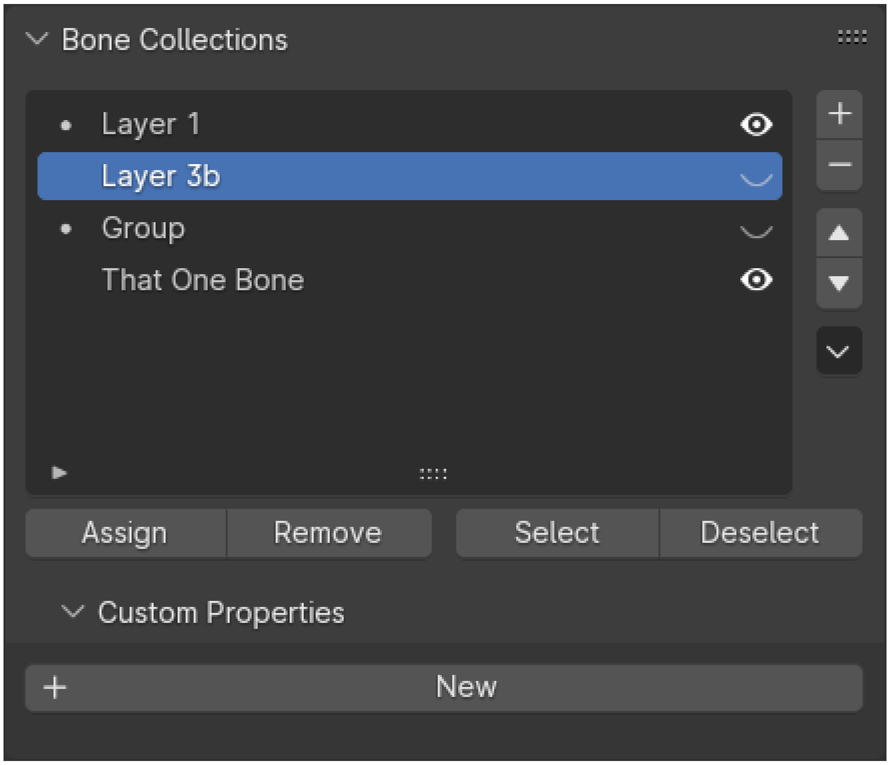
Bone Collections

Note

Bone Collections were introduced in Blender 4.0 as replacement of Armature Layers and Bone Groups. Bone colors are now managed directly on the bone.

Reference

Mode:: Pose & Armature Edit Modes
Panel:: *Properties ‣ Armature ‣ Bone Collections*
Menu:: *Pose ‣ Bone Collections ‣ …*


Bone Collections
Layer 1
Layer 3b
Group
That One Bone
Assign
Remove
Select
Deselect
Custom Properties
New

The Bone Collections panel in the Armature properties.

This panel contains a Tree View to manage Bone Collection From this panel, Bone Collections can be created, deleted, re-arranged, and more.

Collections can be renamed by double clicking on the name, or right clinking and selecting *Rename*. To nest a collection inside an existing collection, click and drag the name onto another collection's name. Child collection can also be made by `RMB` and selecting "Add Child Collection".

To the right of the name gives a few controls of the collection:

Visible (Eye)
: Bones in this collection will be visible in the 3D Viewport.

Solo (Star)
: Show only this bone collection, and others also marked as "solo".

Further more, collection that are not empty will have a dot to indicate the collection has bones assigned to it.

Tip

The Bone Properties panel gives a slightly different view on the bone's collections. See Bone Relations.

Specials

Show All
: Unhides any hidden bone collections.

Un-Solo All
: Clear the 'solo' setting on all bone collections

Remove Unused
: Remove all bone collections that have neither bones nor children. This is done recursively, so bone collections that only have unused children are also removed.

Assign & Select

Assign
  Assigns the selected bones to the active bone collection.

Remove
  Removes the selected bones from the active bone collection.

Select
  Selects the bones in the active bone collection.

Deselect
  Deselects the bones in the active bone collection.

Note

Individual bones can als be unassigned from their collections via the Bone Relations panel.

Tip

For setting up custom selection sets of bones, take a look at the *Selection Sets* add-on. It is bundled with Blender.

Moving Bones between Collections

Blender should be in *Edit Mode* or *Pose Mode* to move bones between collections. Note that as with objects, bones can be assigned to in several collections at once.

Move to Bone Collection
  Shows a list of the Armature's *editable* bone collections. Choosing a bone collection unassign the selected bones from all other bone collections, then assigns them to the chosen one.

Available as *Pose* ‣ *Move to Collection* (*Pose Mode*) *Armature* ‣ *Move to Collection* (*Edit Mode*), and `M` (either mode).

Bone Collections

Shows a list of the Armature’s *editable* bone collections. The collections that the active bone is assigned to are prefixed with a `-`, and choosing those will unassign all selected bones from that collection. Similarly, choosing a bone collection prefixed with a `+` will assign all selected bones to that collection.

Available as *Pose* ‣ *Bone Collections* (*Pose Mode*) *Armature* ‣ *Bone Collections* (*Edit Mode*), and `Shift-M` (either mode).

Note

The above operators will only show the *editable* bone collections. When the Armature is linked, its bone collections will be *read-only*. New bone collections can still be added via library overrides; only those will be editable.

See Library Overrides of Bone Collections.

Custom Properties

Create and manage your own properties to store data in the Bone Collection’s data-block. See the Custom Properties page for more information.

Selection Sets

Reference

Mode:: Pose Mode
Panel:: *Armature ‣ Selection Sets*

Selection Sets are a feature that allows the definition of sets of bones for easy selection while animating. The sets can be created in local and linked armature overrides.

Selection Set

A List View listing all selection sets for the selected armature. Here, selection sets can be renamed by double clicking on the name.

To the right of the name is a check box to include that selection set when copying to the clipboard.

Specials

Delete All Sets

Removes all selection sets from the list.

Remove Selected Bones from All Sets

Removes the selected bones from all selection sets.

Copy Selected Set(s)

Copies the selected set to Blender's clipboard.

Paste Selected Set(s)

Pastes a selection set from Blender's clipboard.

Assign

Assigns the selected bones to the active selection set.

Remove

Removes the selected bones to the active selection set.

Select

Selects all the bones in the active selection set.

Deselect

Deselects all the bones in the active selection set.

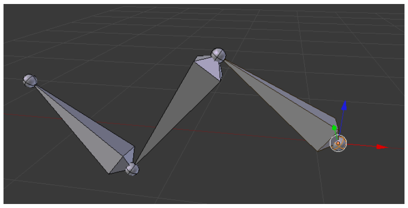
Viewport Display

Reference

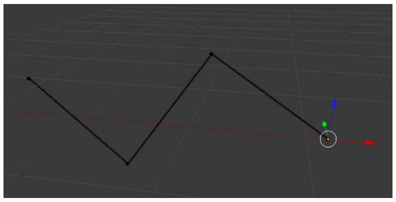
Mode:: All Modes
Panel:: *Armature ‣ Viewport Display*

Display As

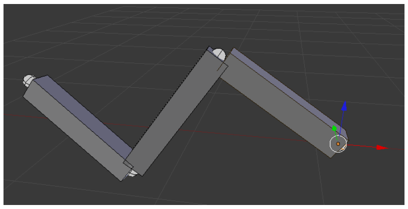
This controls the way the bones appear in the 3D Viewport.



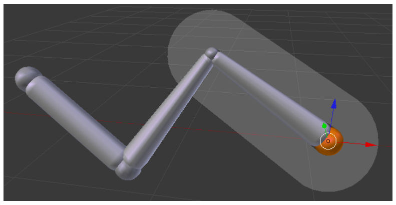
Octahedral bone display.



Stick bone display.



B-Bone bone display.

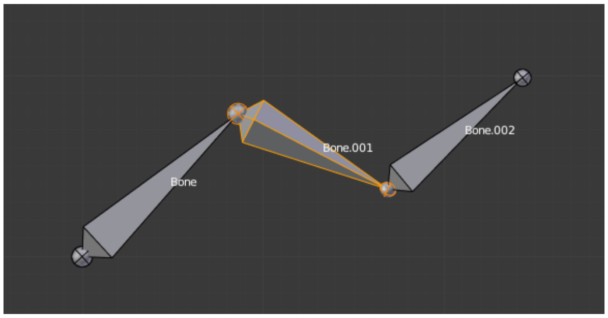


Envelope bone display.

Octahedral

This is the default visualization, well suited for most of editing tasks. It materializes:

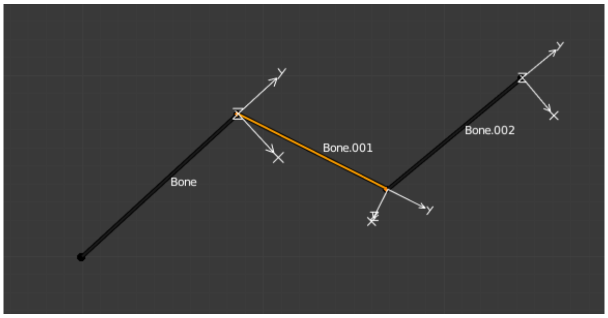
- The bone root ("big" joint) and tip ("small" joint).
- The bone "size" (its thickness is proportional to its length).
- The bone roll (as it has a square section).



Note the 40° rolled Bone.001 bone.

Stick

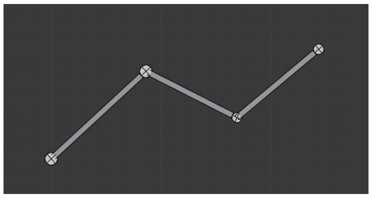
This is the simplest and most non-intrusive visualization. It just materializes bones by sticks of constant (and small) thickness, so it gives you no information about root and tip, nor bone size or roll angle.



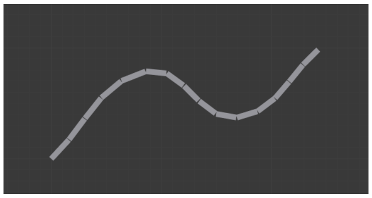
Note that Bone.001 roll angle is not visible (except by its XZ axes).

B-Bone

This visualization shows the curves of “smooth” multi-segmented bones; see the Bendy Bones for details.



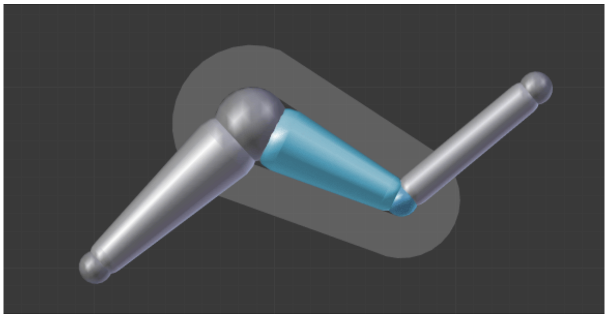
An armature of B-Bones, in Edit Mode.



The same armature in Object Mode.

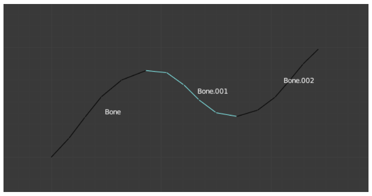
Envelope

This visualization materializes the bone deformation influence. More on this in the bone page.

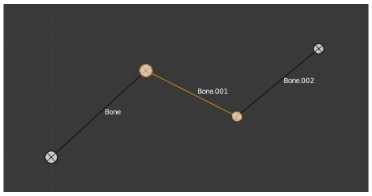


Wire

This simplest visualization shows the curves of “smooth” multi-segmented bones.



An armature of Wire, in Pose Mode.



The same armature in Edit Mode.

Show

Names

Displays the name of each bone.

Shapes

When enabled, the default standard bone shape is replaced, in *Object Mode* and *Pose Mode*, by the shape of a chosen object (see Shaped Bones for details).

Bone Colors

Draws bones in their configured colors. Disable to always draw bones in the default color. For more details see Bone Colors.

In Front

When enabled, the bones of the armature will always be shown on top of the solid objects (meshes, surfaces, …). I.e. they will always be visible and selectable (this is the same option as the one found in the *Display* panel of the *Object data* tab). Very useful when not in *Wireframe* mode.

Axis

When enabled, the (local) axes of each bone are displayed (only relevant for *Edit Mode* and *Pose Mode*).

Position

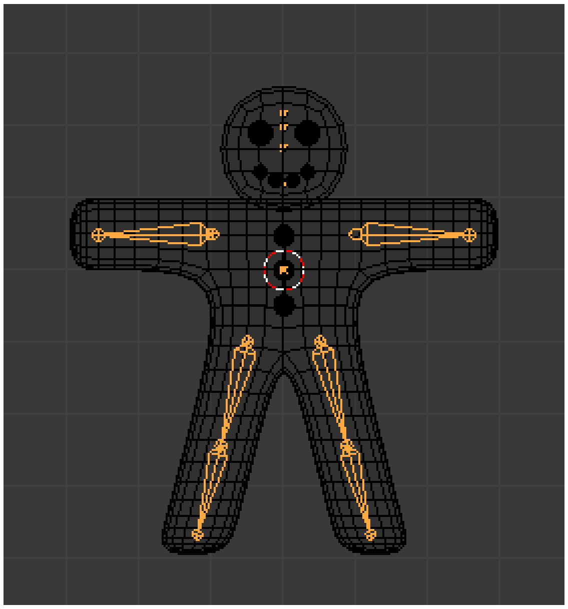
The position for the axes display on the bone. Increasing the value moves it closer to the tip; decreasing moves it closer to the root.

Relations

Whether the Relationship Lines overlay should be drawn from each parent's tail or head. The lines are always drawn towards the childrens' heads.

Structure

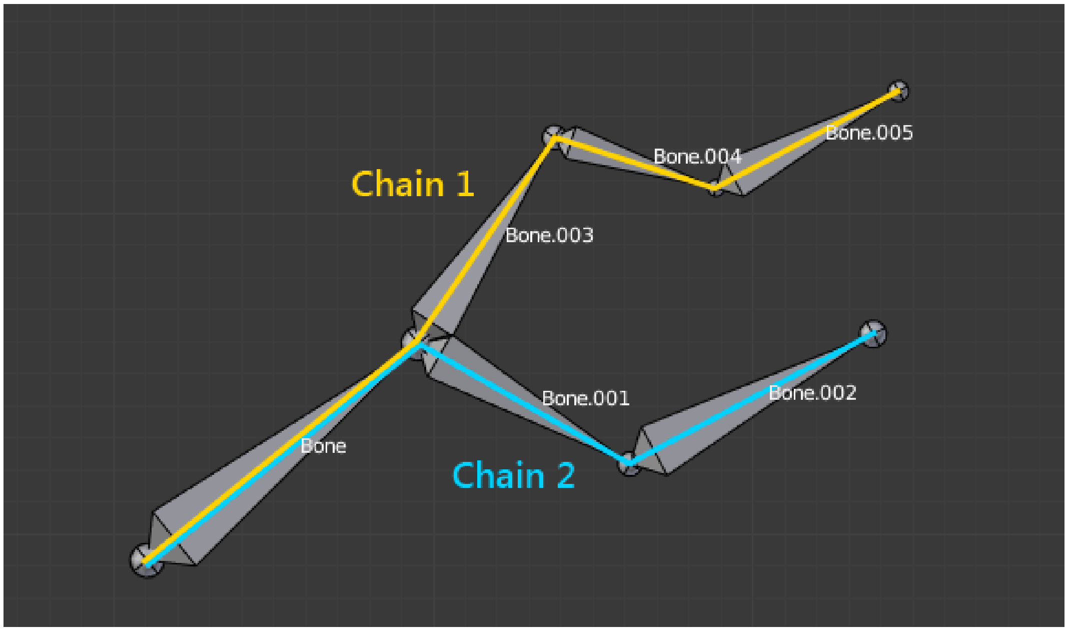
Armatures mimic real skeletons. They are made out of bones, which are (by default) rigid elements. But you have more possibilities than with real skeletons: In addition to the “natural” rotation of bones, you can also move and even scale them! And your bones do not have to be connected to each other; they can be completely free if you want. However, the most natural and useful setups imply that some bones are related to others, forming so-called “chains of bones”, which create some sort of “limbs” in your armature, as detailed in Chains of Bones.



Example of a very basic armature.

Chains of Bones

The bones inside an armature can be completely independent from each other (i.e. the modification of one bone does not affect the others). But this is not often a useful set up: To create a leg, all bones “after” the thigh bone should move “with” it in a well-coordinated manner. This is exactly what happens in armatures by parenting a bone to the next one in the limb, you create a “chains of bones”. These chains can be ramified. For example, five fingers attached to a single “hand” bone.



An armature with two chains of bones.

Bones are chained by linking the tip of the parent to the root of the child. Root and tip can be *connected*, i.e. they are always exactly at the same point; or they can be *free*, like in a standard parent-child object relationship.

A given bone can be the parent of several children, and hence be part of several chains at the same time.

The bone at the beginning of a chain is called its *root bone*, and the last bone of a chain is the *tip bone* (do not confuse them with similar names of bones' joints!).

Chains of bones are a particularly important topic in posing (especially with the standard *forward* kinematics versus "automatic" *inverse* kinematics posing techniques). You create/edit them in *Edit Mode*, but except in case of connected bones, their relationships have no effect on bone transformations in this mode (i.e. transforming a parent bone will not affect its children).

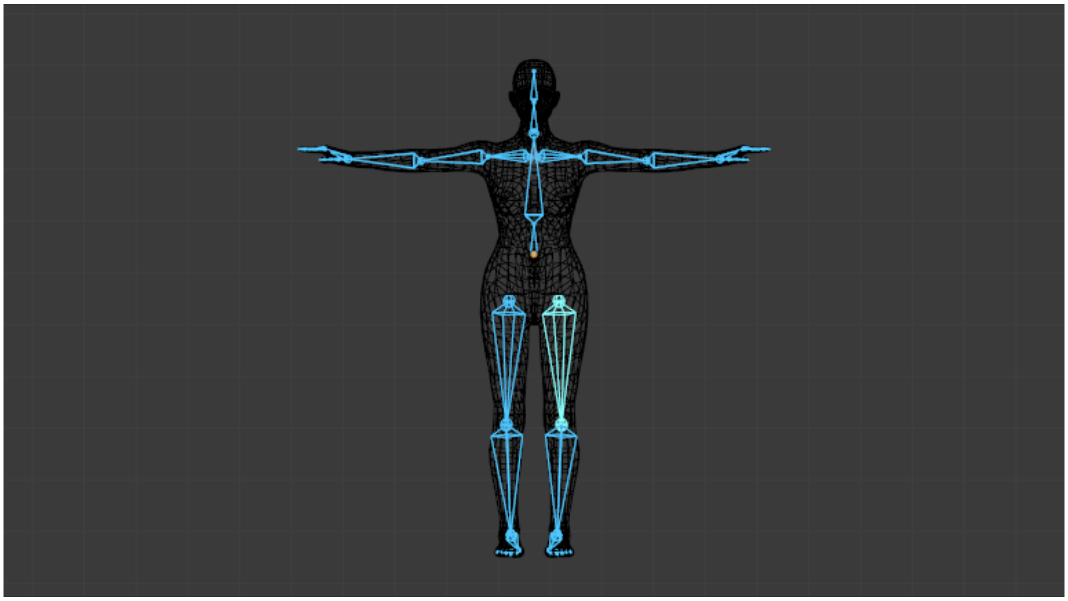
The easiest way to manage bones relationships is to use the Relations panel in the Bone tab.

Skinning

- Introduction
- Armature Deform Parent
    - With Empty Groups
    - With Automatic Weights
    - With Envelope Weights

Introduction

We have seen in previous pages how to design an armature, create chains of bones, etc. Now, having a good rig is not the final goal, unless you want to produce a “Dance Macabre” animation, you will likely want to put some flesh on your skeletons! Surprisingly, “linking” an armature to the object(s) it should transform and/or deform is called the “skinning” process…



The human mesh skinned on its armature.

In Blender, you have two main skinning types:

1. You can Parent/Constrain Objects to Bones – then, when you transform the bones in *Pose Mode*, their “children” objects are also transformed, exactly as with a standard parent/children relationship… The “children” are **never** deformed when using this method.
2. You can Use the Armature Modifier on entire Mesh, and then, some parts of this object to some bones inside this armature. This is the more complex and powerful method, and the only way to really deform the geometry of the object, i.e. to modify its vertices/control points relative positions.

Hint

Retargeting

Retargeting, which is a way to apply motion-capture data (acquired from real world) to a rig, is available through add-ons and importers.

Armature Deform Parent

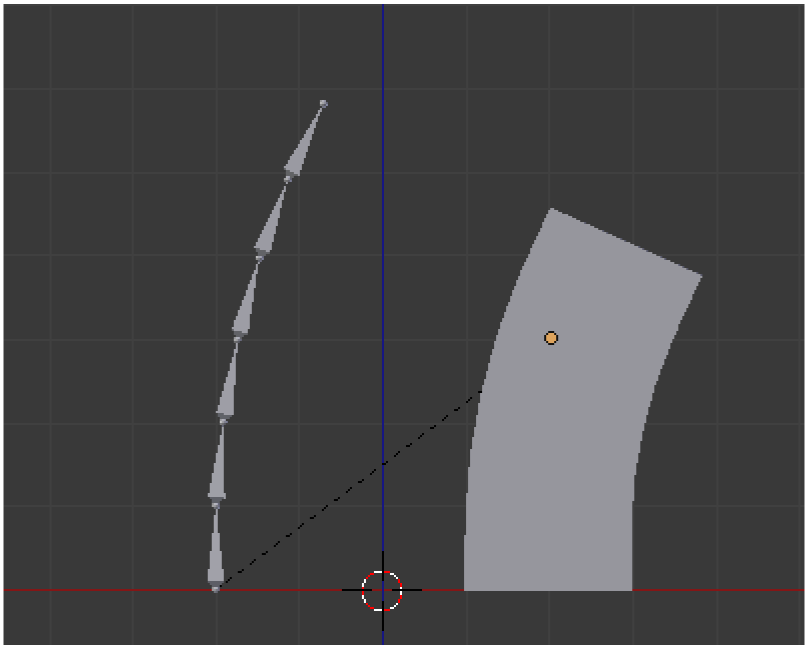
Reference

Mode::	Object Mode and Pose Mode
Menu::	*Object/Pose ‣ Parent ‣ Armature Deform*
Shortcut::	`Ctrl-P`

Armature Deform Parenting is a way of creating and setting up an Armature Modifier.

To use *Armature Deform Parenting* you must first select all the child objects that will be influenced by the armature and then lastly, select the armature object itself. Once all the child objects and the armature are selected, press `Ctrl-P` and select *Armature Deform* in the *Set Parent To* pop-up menu.

The armature will be the parent object of all the other child objects and each child object will have an Armature Modifier with the armature associated (*Object* field).



Bone associated with Mesh Object.

With Empty Groups

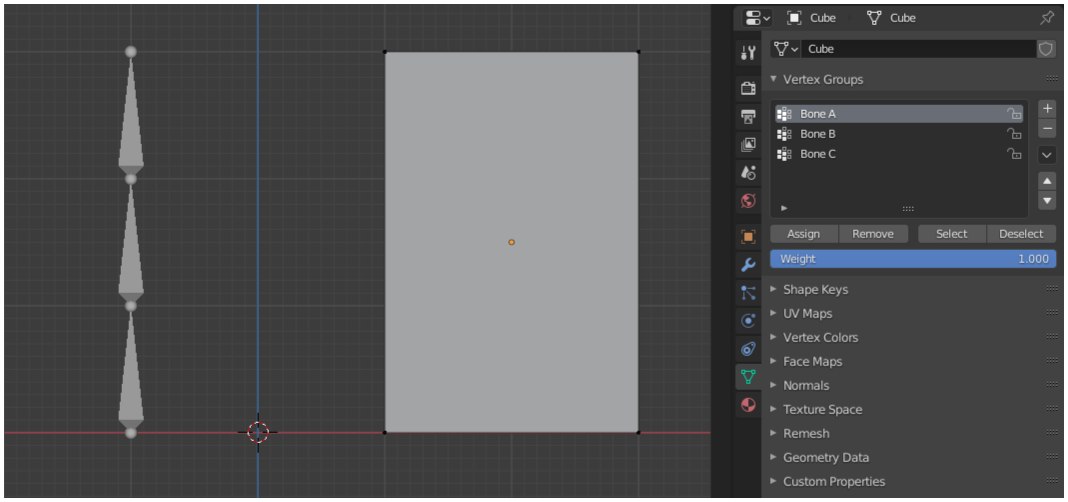
When parenting it will create empty vertex groups on the child objects (if they do not already exist) for and named after each deforming bone in the armature. The newly created vertex groups will be empty. This means they will not have any weights assigned. Vertex groups will only be created for bones which are setup as deforming (*Properties* ▸ *Bone* ▸ *Deform Panel*).

You can then manually select the vertices and assign them to a particular vertex group of your choosing to have bones in the armature influence them.

Choose this option if you have already created (and weighted) all the vertex groups the mesh requires.

Example

For example, if you have an armature which consists of three bones named “BoneA”, “BoneB” and “BoneC” and cube mesh called “Cube”. If you parent the cube to the armature, the cube will get three new vertex groups created on it called “BoneA”, “BoneB” and “BoneC”. Notice that each vertex group is empty.


Cube
Cube
Cube
Vertex Groups
Bone A
Bone B
Bone C
Assign
Remove
Select
Deselect
Weight
1.000
Shape Keys
UV Maps
Vertex Colors
Face Maps
Normals
Texture Space
Remesh
Geometry Data
Custom Properties

Cube in Edit Mode using Armature Deform with empty groups.

With Automatic Weights

With Automatic Weights parenting works similar to *With Empty Groups*, but it will not leave the vertex groups empty. It calculates how much influence a particular bone would have on vertices based on the distance from those vertices to a particular bone ("bone heat" algorithm). This influence will be assigned as weights in the vertex groups.

This method of parenting is certainly easier to setup, but it can often lead to armatures which do not deform child objects in ways you would want. Overlaps can occur when it comes to determining which bones should influence certain vertices when calculating influences for more complex armatures and child objects. Symptoms of this confusion are that when transforming the armature in *Pose Mode*, parts of the child objects do not deform as you expect; If Blender does not give you the results you require, you will have to manually alter the weights of vertices in relation to the vertex groups they belong to and have influence in.

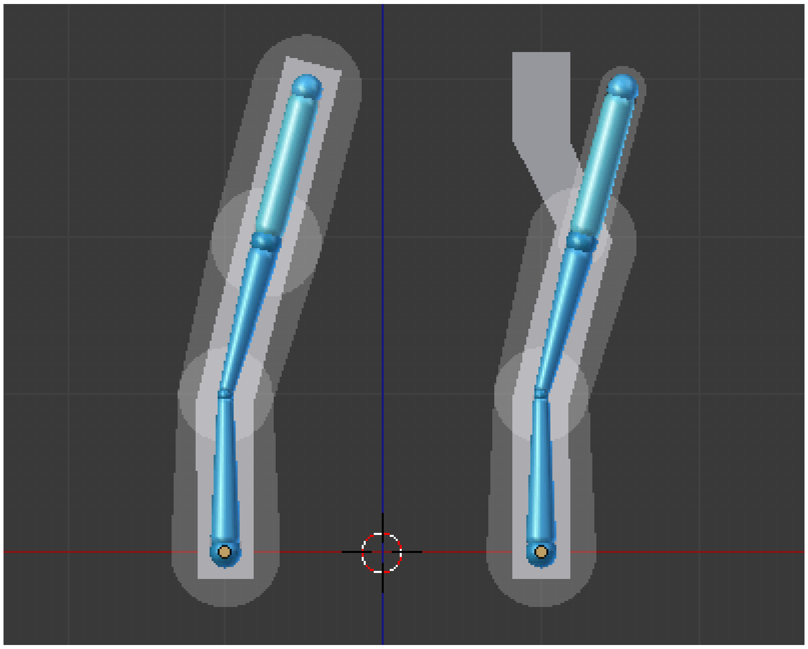
With Envelope Weights

Works in a similar way to *With Automatic Weights*. The difference is that the influences are calculated based on the Bone Envelopes settings. It will assign a weight to each vertex group the vertices that is inside its bone's influence volume, depending on their distance to this bone.

This means newly included/excluded vertices or new envelope settings will not be taken into account. You will have to apply Armature Deform With Envelope Weights parenting again.

Tip

If you want the envelope setting to be used instantly, bind the Armature Modifier to *Bone Envelopes*.



Two sets of armatures, each with three bones.

Warning

If you had defined vertex groups using same names as skinned bones, their content will be completely overridden by both *Automatic* and *Envelope Weights*. In this case *With Empty Groups* could be used instead.

See also

Vertex Groups for Bones.

Posing

- Introduction
    - Visualization
- Selecting
    - All
    - None
    - Invert
    - Box Select
    - Circle Select
    - Lasso Select
    - Select Mirror
    - Select More/Less
    - Select Grouped
    - Select Linked
    - Select Pattern
    - Constraint Target
- Editing
    - Introduction
    - Clear Transform
    - Apply
    - In-Betweens
    - Propagate
    - Copy/Paste Pose
    - Pose Library
    - Flip Quats
    - Show/Hide
- Tool Settings
    - Pose Options
    - Known Limitations
- Bone Constraints
    - Introduction
    - Inverse Kinematics

Introduction

Once an armature is skinned by the needed object(s), you need a way to configure the armature into positions known as poses. Basically, by transforming the bones, you deform or transform the skinned object(s). However, you will notice that you cannot do this in *Edit Mode* – remember that *Edit Mode* is used to edit the default, base, or “rest” position of an armature. You may also notice that you cannot use *Object Mode* either, as here you can only transform whole objects.

So, armatures have a third mode dedicated to the process of posing known as *Pose Mode*. In rest position (as edited in *Edit Mode*), each bone has its own position/rotation/scale to neutral values (i.e. 0.0 for position and rotation, and 1.0 for scale). Hence, when you edit a bone in *Pose Mode*, you create an offset in the transform properties, from its rest position. This may seem quite similar if you have worked with relative shape keys or Delta Transformations.

Even though it might be used for completely static purposes, posing is heavily connected with animation features and techniques. So if you are not familiar at all with animation in Blender, it might be a good idea to read the animation chapter first, and then come back here.

Visualization

Bone State Colors

The color of the bones are based on their state. There are six different color codes, ordered here by precedence (i.e. the bone will be of the color of the bottom-most valid state):

- Gray: Default.
- Blue wire-frame: in Pose Mode.
- Green: with Constraint.

- Yellow: with IK Solver constraint.
- Orange: with Targetless Solver constraint.

Note

When bone colors are enabled, the state colors will be overridden.

Selecting

Selection in *Pose Mode* is very similar to the one in Edit Mode, with a few deviations: You can only select *whole bones* in *Pose Mode*, not roots/tips…

All

Reference

Mode::	Pose Mode
Menu::	*Select ‣ All*
Shortcut::	`A`

Select all selectable bones.

None

Reference

Mode::	Pose Mode
Menu::	*Select ‣ None*
Shortcut::	`Alt-A`

Deselect all bones, but the active bone stays the same.

Invert

Reference

Mode::	Pose Mode
Menu::	*Select ‣ Invert*
Shortcut::	`Ctrl-I`

Toggle the selection state of all visible bones.

Box Select

Reference

Mode::	Pose Mode
Menu::	*Select* ‣ *Box Select*
Shortcut::	`B`

Interactive box selection.

Circle Select

Reference

Mode::	Pose Mode
Menu::	*Select* ‣ *Circle Select*
Shortcut::	`C`

Interactive circle selection.

Lasso Select

Reference

Mode::	Pose Mode
Menu::	*Select* ‣ *Lasso Select*
Shortcut::	`Ctrl-Alt-LMB`

See Select Lasso.

Select Mirror

Reference

Mode::	Pose Mode
Menu::	*Select ‣ Select Mirror*
Shortcut::	`Shift-Ctrl-M`

Flip the selection from one side to another.

Select More/Less

Reference

Mode::	Pose Mode
Menu::	*Select ‣ More/Less*

Parent `[`, Child `]`

You can deselect the active bone and select its immediate parent or one of its children.

Extend Parent `Shift-[`, Extend Child `Shift-]`

Similar to *Parent/Child* but it keeps the active bone in the selection.

Select Grouped

Reference

Mode::	Pose Mode
Menu::	*Select ‣ Select Grouped*
Shortcut::	`Shift-G`

You can select bones, based on various properties, through the *Select Grouped* pop-up menu `Shift-G`:

Collection

Selects all bones that are share at least one bone collection with the active bone.

Color
: Selects all bones that have the same color as the active bone.

Keying Set
: All bones affected by active Keying Set

Select Linked

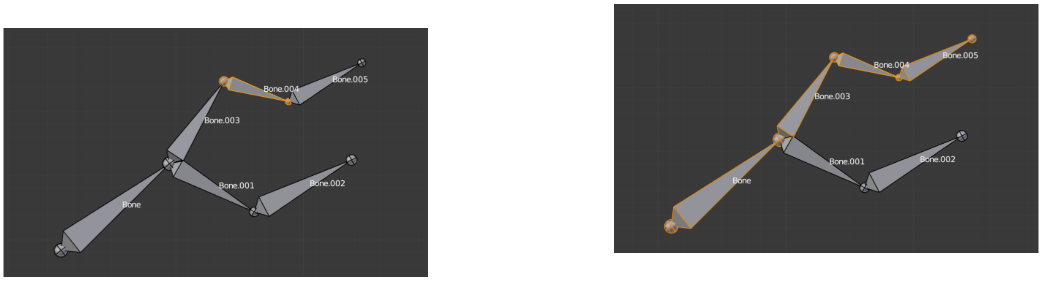
Reference

Mode::	Pose Mode
Menu::	*Select ‣ Select Linked*
Shortcut::	`Ctrl-L`

Selects all the bones in the chain which the active (last selected) bone belongs to.

All Forks
: Selects all bones connected to the active bone even if the branch off from the current bone.

Linked bones selection.


Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone
Bone.004
Bone.005
Bone.003
Bone.001
Bone.002
Bone

A single selected bone.

Its whole chain selected with Linked.

Select Pattern

Reference

Mode::	Pose Mode
Menu::	*Select ‣ Select Pattern…*

Selects all bones whose name matches a given pattern. Supported wildcards: * matches everything, ? matches any single character, [abc] matches characters in “abc”, and [!abc] match any character not in “abc”. As an example *house* matches any name that contains “house”, while floor* matches any name starting with “floor”.

Case Sensitive
    The matching can be chosen to be case sensitive or not.

Extend
    When *Extend* checkbox is checked the selection is extended instead of generating a new one.

Constraint Target

Reference

Mode::	Pose Mode
Menu::	*Select ‣ Constraint Target*

Select bones used as targets for the currently selected bones

Editing

- Introduction
  - Basic Posing
- Clear Transform
- Apply
- In-Betweens
  - Push Pose from Rest Pose
  - Relax Pose to Rest Pose
  - Push Pose from Breakdown
  - Relax Pose to Breakdown
  - Pose Breakdowner
  - Blend to Neighbor
- Propagate
- Copy/Paste Pose
- Pose Library
  - What is a Pose Asset?
  - Creating a Pose Library
  - Using the Pose Library
  - Old Pose Library
- Flip Quats
- Show/Hide

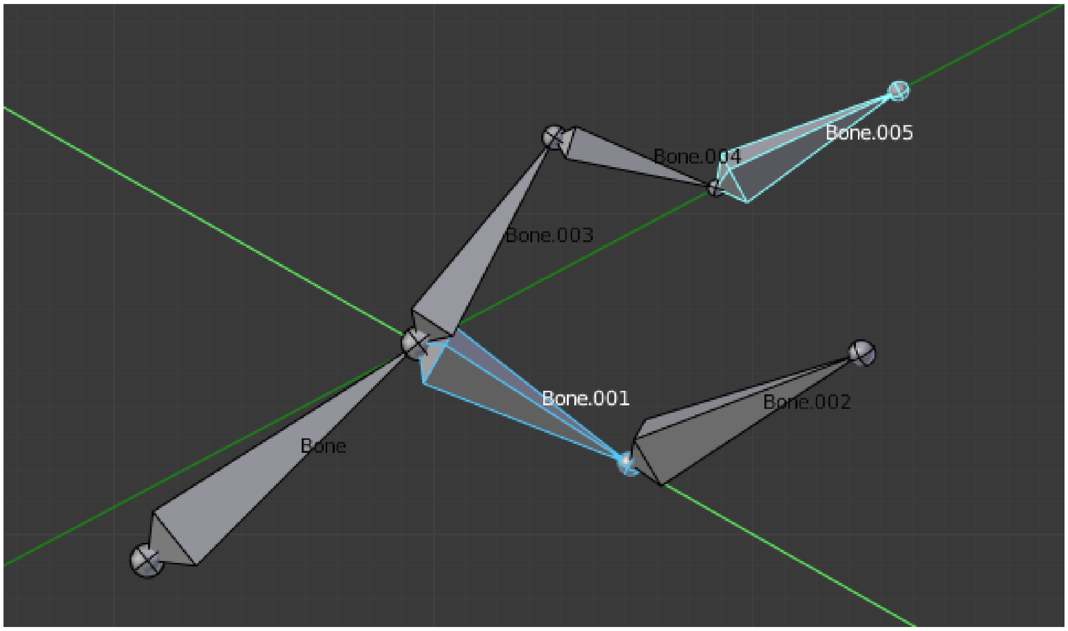
Introduction

In *Pose Mode*, bones behave like objects. So the transform actions (move, rotate, scale, etc.) are very similar to the same ones in Object Mode (all available ones are regrouped in the *Pose ‣ Transform* submenu). However, there are some important specifics:

- Bones' relationships are crucial (see Bone Parenting).
- The "transform center" of a given bone (i.e. its default pivot point, when it is the only selected one) is *its root*. Note by the way that some pivot point options seem to not work properly. In fact, except for the *3D Cursor* one, all others appear to always use the median point of the selection (and not e.g. the active bone's root when *Active Object* is selected, etc.).

Basic Posing

As previously noted, bones' transformations are performed based on the *Rest Position* of the armature, which is its state as defined in *Edit Mode*. This means that in rest position, in *Pose Mode*, each bone has a scale of 1.0, and null rotation and position (as you can see it in the *Transform* panel, in the 3D Viewport's Sidebar).



> An example of a rotation locked to the local Y axis, with two bones selected.
>
> Note that the two green lines materializing the axes are centered on the armature's center, and not each bone's root…

Moreover, the local space for these actions is the bone's own one (visible when you enable the *Axes* option of the *Armature* panel). This is especially important when using axis locking, for example, there is no specific "bone roll" tool in *Pose Mode*, as you can rotate around the bone's main axis just by locking on the local Y axis `R Y Y`… This also works with several bones selected; each one is locked to its own local axis!

When you pose your armature, you are supposed to have one or more objects skinned on it! And obviously, when you transform a bone in *Pose Mode*, its related objects or object's shape is moved/deformed accordingly, in real-time. Unfortunately, if you have a complex rig set-up and/or a heavy skin object, this might produce lag during interactive editing. If you experience such troubles, try enabling the *Delay Deform* button in the *Armature* panel the skin objects will only be updated once you confirm the transform operation.

Clear Transform

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ Clear Transform*

Once you have transformed some bones, if you want to return to their rest position, just clear their transformations.

All

Resets location, rotation, and scaling of selected bones to their default values.

Location, Rotation, Scale `Alt-G`, `Alt-R`, `Alt-S`

Clears individual transforms.

Note that in *Envelope* visualization, `Alt-S` does not clear the scale, but rather scales the *Distance* influence area of the selected bones. (This is also available through the *Pose ‣ Scale Envelope Distance* menu entry, which is only effective in *Envelope* visualization, even though it is always available…)

Reset Unkeyed

Clears the transforms to their keyframe state.

Only Selected

Operate on just the selected or all bones.

Apply

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ Apply*
Shortcut::	`Ctrl-A`

Pose as Rest Pose

Conversely, you may define the current pose as the new rest pose (i.e. "apply" current transformations to the *Edit Mode*). When you do so, the skinned objects/geometry is **also** reset to its default, undeformed state, which generally means you will have to skin it again.

Pose Selected as Rest Pose

Same as *Pose as Rest Pose* but only applies to selected bones.

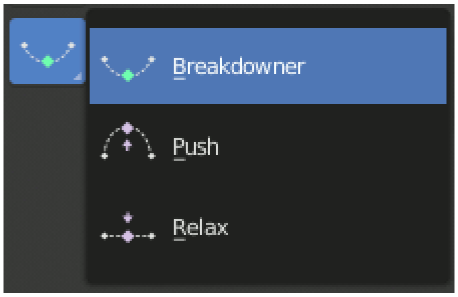
Visual Transform to Pose

Applies the position of the bone after Constraints; allowing the constraints to be deleted and the bones will remain in their constrained positions.

Assign Custom Property Values as Default

Assign the current values of custom properties as their defaults, for use as part of the rest pose state in NLA track mixing.

In-Betweens



In-Betweens Tools.

There are several tools for editing poses in an animation.

There are also in *Pose Mode* a bunch of armature-specific editing options/tools, like auto-bones naming, properties switching/enabling/disabling, etc., that were already described in the armature editing pages. See the links above…

Push Pose from Rest Pose

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ In-Betweens ‣ Push Pose from Rest Pose*

Similar to *Push Pose from Breakdown* but interpolates the pose to the rest position instead. Only one keyframe is needed for this tool unlike two for the other.

Relax Pose to Rest Pose

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ In-Betweens ‣ Relax Pose to Rest Pose*

Similar to *Relax Pose to Breakdown* but works to bring the pose back to the rest position instead. Only one keyframe is needed for this tool unlike two for the other.

Push Pose from Breakdown

Reference

Mode::	Pose Mode
Tool::	*Toolbar ‣ In-Betweens Tools ‣ Push*
Menu::	*Pose ‣ In-Betweens ‣ Push Pose from Breakdown*
Shortcut::	`Ctrl-E`

Push Pose interpolates the current pose by making it closer to the next keyframed position.

Relax Pose to Breakdown

Reference

Mode::	Pose Mode
Tool::	*Toolbar ‣ In-Betweens Tools ‣ Relax*
Menu::	*Pose ‣ In-Betweens ‣ Relax Pose to Breakdown*
Shortcut::	`Alt-E`

Relax pose is somewhat related to the above topic, but it is only useful with keyframed bones. When you edit such a bone (and hence take it "away" from its "keyed position"), using this tool will progressively "bring it back" to its "keyed position", with smaller and smaller steps as it comes near it.

Pose Breakdowner

Reference

Mode::	Pose Mode
Tool::	*Toolbar region ‣ In-Betweens Tools ‣ Breakdowner*
Menu::	*Pose ‣ In-Betweens ‣ Pose Breakdowner*
Shortcut::	`LMB`-drag

Creates a suitable breakdown pose on the current frame.

The Breakdowner tool can be constrained to work on specific transforms and axes, by pressing the following keys while the tool is active:

- `G`, `R`, `S`: move, rotate, scale
- `B`: Bendy bones
- `C`: custom properties
- `X`, `Y`, `Z`: to the corresponding axes

Blend to Neighbor

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ In-Betweens ‣ Blend to Neighbor*
Shortcut::	`Shift-Alt-E`

Transitions the current pose with the neighboring keyframes in the timeline. In order for this operator to work, there must be a keyframe before and after the current frame.

Propagate

Reference

Mode::	Pose Mode
Menu::	*Pose* ‣ *Propagate*
Shortcut::	`Alt-P`

The Propagate tool copies the pose of the selected bones on the current frame over to the keyframes delimited by the *Termination Mode*. It automates the process of copying and pasting.

Termination Mode

Modes which determine how it decides when to stop overwriting keyframes.

To Next Keyframe

Simply copies the pose to the first keyframe after (but not including any keyframe on) the current frame.

To Last Keyframe

Will simply replace the last keyframe (i.e. making action cyclic).

Before Frame

To all keyframes between current frame and the *End frame* option. This option is best suited for use from scripts due to the difficulties in setting this frame value, though it is possible to set this manually via the Adjust Last Operation panel if necessary.

Before Last Keyframe

To all keyframes from current frame until no more are found.

On Selected Keyframes

Will apply the pose of the selected bones to all selected keyframes.

On Selected Markers

To all keyframes occurring on frames with Scene Markers after the current frame.

End Frame

Defines the upper-bound for the frame range within which keyframes will be affected (with the lower bound being the current frame).

Copy/Paste Pose

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ Copy Pose*, *Pose ‣ Paste Pose*, *Pose ‣ Paste Pose Flipped*
Shortcut::	`Ctrl-C`, `Ctrl-V`, `Shift-Ctrl-V`

Blender allows you to copy and paste a pose, either through the *Pose* menu, or by using hotkeys.

Copy Pose
: Copy the current pose of selected bones into the pose buffer.

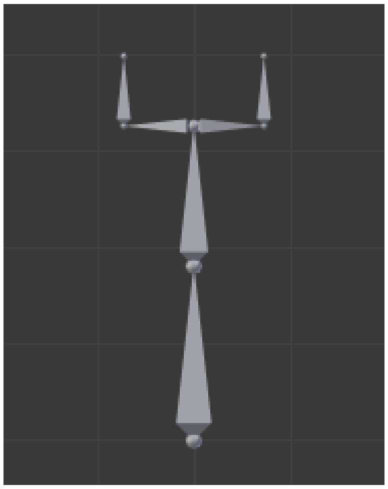
Paste Pose
: Paste the buffered pose to the currently posed armature.

Paste Pose Flipped
: Paste the *X axis mirrored* buffered pose to the currently posed armature.

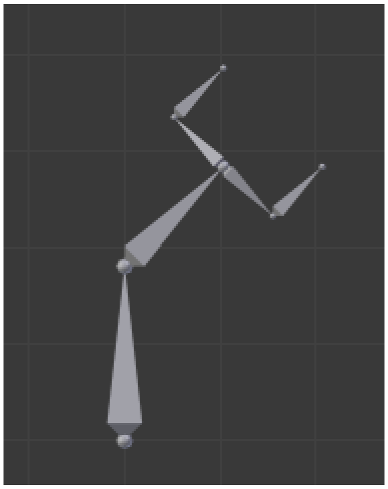
Here are important points:

- This tool works at the Blender session level, which means you can use it across armatures, scenes, and even files. However, the pose buffer is not saved, so you lose it when you close Blender.
- There is only one pose buffer.
- Only the selected bones are taken into account during copying (i.e. you copy only selected bones' pose).
- During pasting, on the other hand, bone selection has no importance. The copied pose is applied on a per-name basis (i.e. if you had a `forearm` bone selected when you copied the pose, the `forearm` bone of the current posed armature will get its pose when you paste it – and if there is no such named bone, nothing will happen…).

- What is copied and pasted is in fact the position, rotation or scale of each bone, in its own space. This means that the resulting pasted pose might be very different from the originally copied one, depending on:
    - The rest position of the bones.
    - And the current pose of their parents.

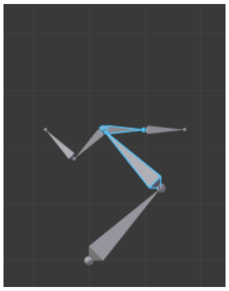


The rest position of the original armature.

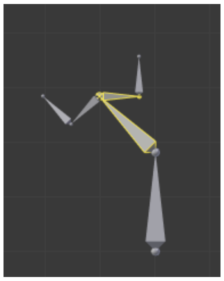


The rest position of the destination armature.

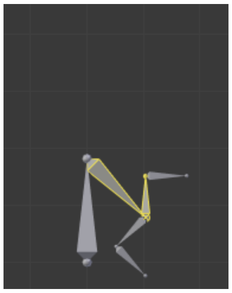
Examples of pose copy/paste.



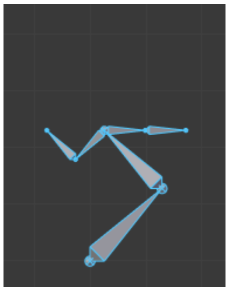
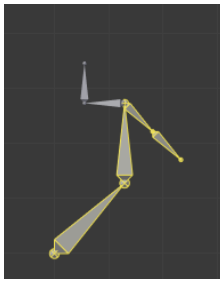
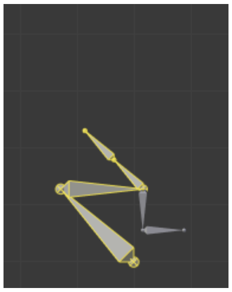
The first copied pose (note that only two bones are selected and



The pose pasted on the destination armature.



The pose mirror-pasted on the destination armature.

hence copied).		
 The same pose as above is copied, but this time with all bones selected.	 The pose pasted on the destination armature.	 The pose mirror-pasted on the destination armature.

Pose Library

This section describes the pose library, which is based on the Asset Browser. For an overview of the asset system, see the Asset Libraries section. The pose library is meant to be used in Pose Mode. In other words, it only works when posing an armature, and not for general object animation.

Note

The pose library is implemented as an add-on. This add-on is enabled by default; disabling it will remove the pose library from Blender's user interface.

The "building blocks" of the pose library are actually implemented in Blender itself. The add-on only contains the user interface and the logic that determines what is stored in a pose asset. This was intentionally put into an add-on, so that artists or studios who want to change the behavior can do so with an add-on of their own.

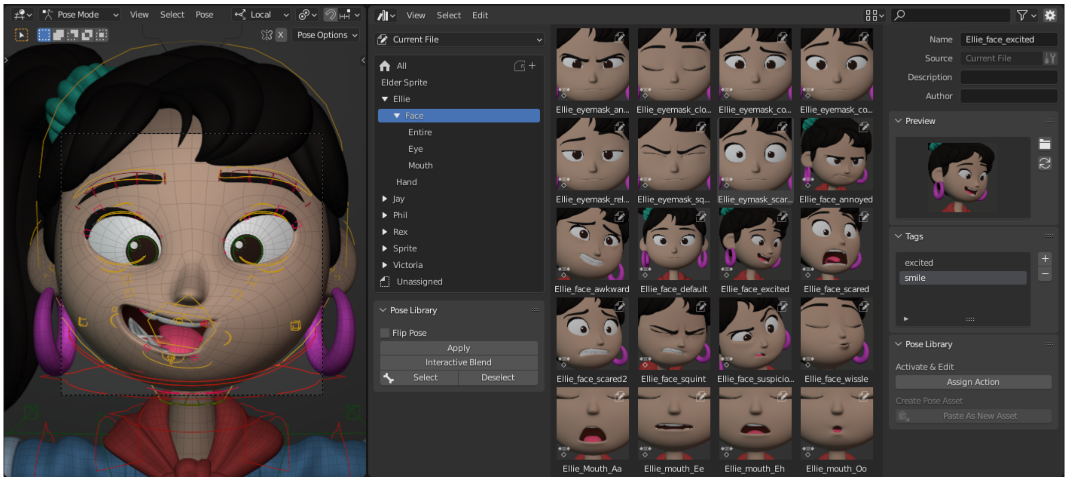
What is a Pose Asset?

A *pose asset* is an action that has been marked as asset, and that contains exactly **one frame of animation** data. Usually these are created via the *Create Pose Asset* button (see below), but any action that is keyed on exactly one frame can be seen as pose asset.

Each pose in the library is stored in its own action data-block. This means that it can get its own name, its own preview image, and can be organized in Asset Catalogs.

Creating a Pose Library

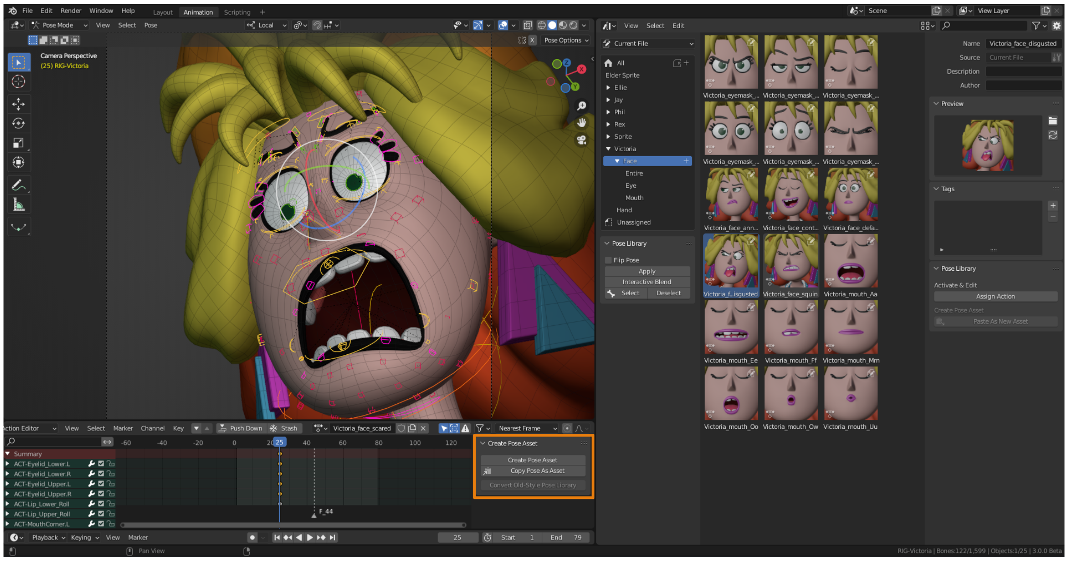
A *pose library file* is typically a blend-file that is dedicated to poses. It can link in a character, props, etc., which can then not only be used to create the poses, but also for rendering previews.


Pose Mode
View
Select
Pose
Local
Pose Options
Current File
All
Elder Sprite
Ellie
Face
Entire
Eye
Mouth
Hand
Jay
Phil
Rex
Sprite
Victoria
Unassigned
Pose Library
Flip Pose
Apply
Interactive Blend
Select
Deselect
Ellie_eyemask_an...
Ellie_eyemask_clo...
Ellie_eyemask_co...
Ellie_eyemask_co...
Ellie_eyemask_rel...
Ellie_eyemask_sq...
Ellie_eymask_scar...
Ellie_face_annoyed
Ellie_face_awkward
Ellie_face_default
Ellie_face_excited
Ellie_face_scared
Ellie_face_scared2
Ellie_face_squint
Ellie_face_suspicio...
Ellie_face_wissle
Ellie_Mouth_Aa
Ellie_mouth_Ee
Ellie_mouth_Eh
Ellie_mouth_Oo
Name
Ellie_face_excited
Source
Current File
Description
Author
Preview
Tags
excited
smile
Pose Library
Activate & Edit
Assign Action
Create Pose Asset
Paste As New Asset

Example pose library of the Sprite Fright character Ellie.

Pose Creation via Action Editor

To create a pose in the library from the Action Editor, **pose the character, select the relevant bones**, and click the *Create Pose Asset* button. This will create the new pose Action, which will contain keys for the current value of each bone’s location, rotation, scale, and Bendy Bone properties.


File
Edit
Render
Window
Help
Layout
Animation
Scripting
Pose Mode
Camera Perspective
(25) RIG-Victoria
Action Editor
Push Down
Stash
Victoria_face_scared
Nearest Frame
Summary
ACT-Eyelid_Lower.L
ACT-Eyelid_Lower.R
ACT-Eyelid_Upper.L
ACT-Eyelid_Upper.R
ACT-Lip_Lower_Roll
ACT-Lip_Upper_Roll
ACT-MouthCorner.L
Create Pose Asset
Create Pose Asset
Copy Pose As Asset
Convert Old-Style Pose Library
F_44
Playback
Keying
View
Marker
Start 1
End 79
Current File
All
Elder Sprite
Ellie
Jay
Phil
Rex
Sprite
Victoria
Face
Entire
Eye
Mouth
Hand
Unassigned
Pose Library
Flip Pose
Apply
Interactive Blend
Select
Deselect
Victoria_face_disgusted
Victoria_face_squint
Victoria_mouth_Aa
Victoria_mouth_Ee
Victoria_mouth_Ff
Victoria_mouth_Mm
Victoria_mouth_Oo
Victoria_mouth_Ow
Victoria_mouth_Uu
Name
Victoria_face_disgusted
Source
Current File
Description
Author
Preview
Tags
Pose Library
Activate & Edit
Assign Action
Create Pose Asset
Paste As New Asset
Scene
View Layer
RIG-Victoria | Bones:122/1,599 | Objects:1/25 | 3.0.0 Beta

> To create a new pose asset, use the Create Pose Asset button in the Action editor.

The created Action is now assigned to the armature. This makes it possible to inspect which bones are included and to tweak anything. In that respect, it’s an Action like any other, and you can add or remove keys as usual. Just make sure that the keys are all on the same frame, in order to keep this a “pose” instead of an “animation snippet”; the latter isn’t supported at the moment.

True to its name, the *Create Pose Asset* button automatically marks the Action as Asset. Not only does this make it available in the pose library, it will also act as a *fake user* to ensure the Action isn’t lost after you unassign it from the armature.

The pose asset can be renamed in the Asset Browser. There you can also right click on the thumbnail, then choose Assign Action to assign the Action to the active Object (see description above).

Note

The **Create Pose Asset** button creates a new Action. To make sure that this is actually visible in the user interface, so that you know that something happened, it tries to make sure that the Asset Browser shows the newly created pose asset. Because of this, it **requires that there is an Asset Browser visible, and that it’s set to show the Current File asset library**.

This is especially relevant to pose assets, compared to other assets. You cannot mark an object as asset multiple times, but you can create ten pose assets from the same character pose.

Pose Creation from Existing Animation

Animators eat and breathe time, so there is a fair chance that you already have some poses lined up on the timeline. Creating a pose asset from existing animation is pretty much the same as described above, with a few subtle differences:

- Go to the frame with the pose you want to turn into an asset.
- Select the relevant bones and click the **Create Pose Asset button** in the Action Editor.
- This creates an Action as before, but this time **it also includes any bone property that was keyed on the current frame**. In other words: any bone property (regular and custom) that's displayed in yellow in the user interface will be included in the pose asset. This makes it possible to also include properties that control IK/FK switching, for example. As with the pose, the current value is copied into the pose asset, and not the keyed value.
- Blender saves which action was previously assigned to the armature.
- The new pose Action is assigned to the armature so you can give it a name and inspect/adjust its contents.
- Click the *Restore Previous Action* button (back arrow icon) that appeared next to the Create Pose Asset button. This **reassigns the previous Action**, so that you're back at the animation you had before.

Pose Creation by Copying from Other File

As described in Design Limitations, Blender only writes data to the currently open blend-file. To copy a pose from some other file into a pose library file, see the following steps:

- Pose the character and select the relevant bones.
- Click the **Copy Pose as Asset button**, which is available in the Action Editor. This will create the pose asset (including its thumbnail) and store it in a temporary file somewhere.
- Choose an existing pose asset, and open its context menu. Click the **Open Blend File** option.
- A new Blender process will start, and automatically open the asset library file that contains the chosen pose. By the way, this works for all assets, not just poses!
- In the Asset Browser, click the **Paste as New Asset button**. This will load that temporary file, and load all the assets it can find in there. In

our case, it will only find a single pose, but future versions of Blender may extend this for other asset types. This is why the button is named so generically – it is not pose-specific.

- Give the pose a name, and click on the “refresh” button in the preview image panel to render a new preview if you want.
- **Save the file and quit Blender**.
- The original Blender is still running in the background and notices that the new Blender has quit. It **automatically refreshes the Asset Browser** to show the newly added pose.

Automatically Assigned Catalog

When you create a pose asset, Blender may automatically assign it to an asset catalog. This only works if there is an Asset Browser visible; Blender then assigns the pose asset to its active asset catalog. If there are multiple Asset Browsers open, it performs the following steps:

- If the current window has one Asset Browser, it uses that one.
- If the current window has multiple Asset Browsers, it uses the biggest one.
- Otherwise Blender goes over the other windows (if there are any), and do a similar search. The first window it sees that has an Asset Browser wins.

Controlling the Look of Preview Images

The pose library preview images are rendered with the active Scene camera. This approach was preferred over rendering a specific 3D Viewport for two main reasons:

- There is only one scene camera active at any time, making it predictable which camera is used.
- The camera, as well as the rest of the scene, can be set up specifically for rendering the thumbnails. Pose library files are intended for that purpose: to contain the poses and render their preview images.

The preview images are rendered using the Workbench Engine. Switch the scene to use that as render engine, and you'll see various options to influence the look. Select a pose asset and press the Generate Preview button to re-render the preview image with the current settings.

You can also animate settings such as MatCap rendering, light positions, and intensities, etc. Use this to your advantage!

Scene Animation for Preview Images

Sometimes it's handy to have a few different background colors or camera angles for your poses. Many facial poses are made with a specific camera angle in mind.

- Background color can be animated by placing a plane behind the character and animating its material. In this case just for fun, but for more serious applications this could be used to indicate a certain character, or a mood, or anything else.
- The active camera can be switched by using camera markers.

Both make it possible to choose a specific frame to pick the background color and camera angle. Pose the character, click the *Create Pose Asset* button, and the pose action will be keyed on the current frame. This means it's easy to edit the pose and refresh its preview image, because you know exactly which frame it was originally created on.

Using the Pose Library

The pose library can be used to pose a character in a few different ways. In short, you can fully apply a pose or blend it into the character's current pose interactively. How exactly these operations work depends on where you use them. This section will explain the use from both the Asset Browser and the 3D Viewport.

Use from the Asset Browser

The pose library can be used directly from the Asset Browser. The **Pose Library panels will appear** when the active object is an armature and in Pose Mode. The catalog system and the filter bar at the top can be used to search for specific poses.

The following operators can be accessed by `RMB` on a pose:

Apply Pose

Applies the pose to the character. If there are any bones selected, the pose will be applied only to those bones. This makes it possible to create a "finger guns" pose by applying a fist pose to the hand, and then an "open hand" pose for only the index finger and thumb. Double-clicking a pose will also apply it.

Apply Pose Flipped

Will mirror the pose from left to right and vice versa. This makes it possible, for example, to apply a left-hand pose to the right hand, reducing the number of poses you have to put into the library. This can of course also be applied for asymmetrical facial expressions that depend on the camera angle. While blending (see below), keep `Ctrl` pressed to blend the flipped pose.

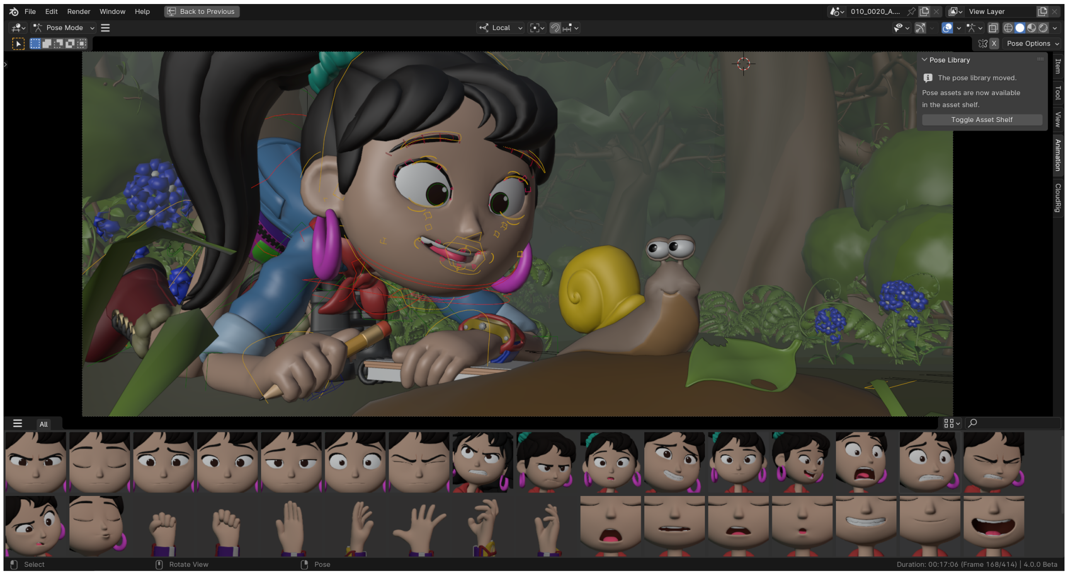
Blend Pose

Allows you to gradually blend a pose from the library into the character's pose. Click the button, then move the mouse left/right to determine the desired blend. A pose asset can be "subtracted" while blending. Drag to the right to blend as usual, drag to the left to subtract the pose. While blending, you can use `Tab` to toggle between the original and the blended pose. As usual in Blender, `LMB` or press `Return` to confirm; `RMB` or press `Esc` to cancel the operator. Blending can also exaggerate a pose, by pressing `E` (for Extrapolate) and applying a pose for more than 100%.

Select/Deselect Pose Bones

Select or deselect the bones that are used in the pose. This can be used to create a selection set, or simply show what was part of the pose and what wasn't.

Use from 3D Viewport


File
Edit
Render
Window
Help
Back to Previous
Pose Mode
Local
View Layer
Pose Options
Pose Library
Toggle Asset Shelf
Item
Tool
View
Animation
CloudRig
All
Select
Rotate View
Pose
Duration: 00:17:06 (Frame 168/414) | 4.0.0 Beta

The pose library moved.

Pose assets are now available in the asset shelf.

The pose library in use from the Asset Shelf.

Note

The pose library previously lived in the Sidebar within the Pose Library panel. The panel still exists, but now contains a button to open the asset shelf.

In the 3D viewport, poses can be quickly applied from the Asset Shelf. Contrary to the Asset Browser, the shelf allows you to apply poses quicker.

Click on a pose to apply it. A single click is enough. You can also select and apply a pose via the cursor keys. This allows for fast exploration of the poses, to directly see the result on the active character.

Drag the pose thumbnail left to right to blend it into the character's current pose. Just release the mouse button to confirm.

Old Pose Library

In Blender 3.0, the Asset Browser based pose library, described above, replaced its predecessor pose library system. This section describes how to convert poses from the old pose library to the current system.

Converting Old Pose Libraries

Old-style pose libraries can be converted to pose assets in the following way:

- In the Action Editor, select the Action containing the pose library you want to convert.
- Make sure the scene camera is set up correctly for rendering preview images.
- In the Action Editor’s Pose Library panel, click the “Convert Old-Style Pose Library” button.
- Open the Asset Browser, and see the poses have been converted.
- If you’re happy with the result, remove the old pose library Action.
- Save the blend-file.

As usual, the blend-file should be saved to a directory marked as asset library in order to use the pose assets from other blend-files.

Note

This conversion does not assign the poses to any catalog, and so they will appear in the “Unassigned” section of the “Current File” asset library.

Flip Quats

Reference

Mode::	Pose Mode
Menu::	*Pose ‣ Flip Quats*
Shortcut::	`Alt-F`

Flip quaternion values to achieve desired rotations, while maintaining the same orientations.

Show/Hide

Reference

Mode::	All Modes
Panel::	*Properties ‣ Bone ‣ Viewport Display*
Menu::	*… ‣ Show/Hide*

You do not have to use bone layers to show/hide some bones. As with objects, vertices or control points, you can use `H`:

- `H` will hide the selected bone(s).
- `Shift-H` will hide all bones *but the selected one(s)*.
- `Alt-H` will show all hidden bones.

You can also use the *Hide* checkbox of the *Bone tab ‣ Viewport Display panel*.

Note that hidden bones are specific to a mode, i.e. you can hide some bones in *Edit Mode*, they will still be visible in *Pose Mode*, and vice versa. Hidden bones in *Pose Mode* are also invisible in *Object Mode*. And in *Edit Mode*, the bone to hide must be fully selected, not just its root or tip.

Tool Settings

Pose Options

Auto IK

Reference

Mode::	Pose Mode
Panel::	*Sidebar ‣ Tool ‣ Pose Options ‣ Auto IK*

Automatic IK is a tool for quick posing, when enabled, translating a bone will activate inverse kinematics and rotate the parent bone, and the parent's parent, and so on, to follow the selected bone. The IK chain can only extend from a child to a parent bone if the child is *connected* to it.

While moving bones, the length of the chain (the number of affected bones) can be increased or decreased using keyboard hotkeys. Pressing `PageUp` will increase in chain length by one and `PageDown` decreases the length by one. The chain length can also be controlled with `WheelUp` or `WheelDown`.

The initial chain length is 0, which effectively means follow the connections to parent bones as far as possible, with no length limit. So pressing increasing the chain length the first time sets the length to 1 (move only the selected bone), and at this point, decreasing the length point sets it back to 0 (unlimited) again. Thus, you have to increase the chain length *more than once* from the initial state to set a finite chain length greater than 1.

This is a more limited feature than using an IK constraint, which can be configured, but it can be useful for quick posing.

X-Axis Mirror

Reference

Mode::	Edit and Pose Mode
Panel::	*Sidebar ‣ Tool ‣ Options ‣ X-Axis Mirror*

This option enables automatic mirroring of editing actions along the X axis. You can enable this option in the *Tool tab ‣ Options panel*, while the armature is selected in *Edit Mode*. When you have pairs of bones of the same name with just a different "side suffix" (e.g. ".R"/".L", or "_right"/"_left" …), once this option is enabled, each time you transform (move, rotate, scale…) a bone, its "other side" counterpart will be transformed accordingly, through a symmetry along the armature local X axis. As most rigs have at least one axis of symmetry (animals, humans, …), it is an easy way to keep the model symmetrical.

Relative Mirror

Reference

Mode::	Edit and Pose Mode
Panel::	*Sidebar ‣ Tool ‣ Options ‣ Relative Mirror*

Accounts for any relative transformations when using *X-Axis Mirror*.

See also

Naming bones.

Known Limitations

Relative Mirror is not supported with Auto IK enabled.

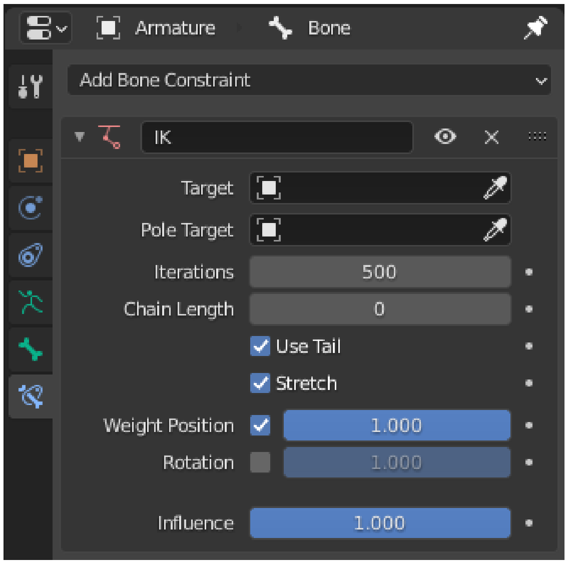
Bone Constraints

- Introduction

Inverse Kinematics

- Introduction
- Spline IK

Introduction



Target
Pole Target
Iterations 500
Chain Length 0
☑ Use Tail
☑ Stretch
Weight Position ☑ 1.000
Rotation ☐ 1.000
Influence 1.000

The Bone Constraints Properties in Pose Mode, with an Inverse Kinematics constraint added to the active bone.

As bones behave like objects in *Pose Mode*, they can also be constrained. This is why the *Constraints* tab is shown in both *Object Mode* and *Edit Mode*. This panel contains the constraints *of the active bone* (its name is displayed at the top of the panel, in the *To Bone:…* static text field).

Constraining bones can be used to control their degree of freedom in their pose transformations, using e.g. the *Limit* constraints. You can also use constraints to make a bone track another object/bone (inside the same object, or in another armature), etc. And the inverse kinematics feature is also mainly available through the *IK Solver* constraint, which is specific to bones.

For example, a human elbow cannot rotate backward (unless the character has broken their arm), nor to the sides, and its forward and roll rotations are limited in a given range. (E.g. depending on the rest position of your elbow, it may be from (0 to 160) or from (-45 to 135).)

So you should apply a *Limit Rotation* constraint to the forearm bone (as the elbow movement is the result of rotating the forearm bone around its root).

Using bones in constraints, either as owners or as targets, is discussed in detail in the constraints pages.

Introduction

Inverse Kinematics (IK) simplifies the animation process, and makes it possible to make more advanced animations with lesser effort.

Inverse Kinematics allow you to position the last bone in a bone chain and the other bones are positioned automatically. This is like how moving someone's finger would cause their arm to follow it. By normal posing techniques, you would have to start from the root bone, and set bones sequentially until you reach the tip bone: When each parent bone is moved, its child bone would inherit its location and rotation. Thus making tiny precise changes in poses becomes harder farther down the chain, as you may have to adjust all the parent bones first.

This effort is effectively avoided by use of IK.

IK is mostly done with bone constraints although there is also a simple Auto IK feature in Pose Mode. They work by the same method but the constraints offer more options and control. Please refer to the following pages for details about these constraints:

- IK Solver
- Spline IK

Armature IK Panel

Reference

Mode::	Pose Mode
Panel::	*Properties* ▸ *Armature* ▸ *Inverse Kinematics*

This panel is used to select the IK Solver type for the armature: *Standard* or *iTaSC*. Most the time people will use the *Standard* IK solver.

Standard

TODO.

iTaSC

iTaSC stands for instantaneous Task Specification using Constraints.

iTaSC uses a different method to compute the Jacobian, which makes it able to handle other constraints than just end effectors position and orientation: iTaSC is a generic multi-constraint IK solver. However, this capability is not yet fully exploited in the current implementation, only two other types of constraints can be handled: Distance in the Cartesian space, and Joint Rotation in the joint space. The first one allows maintaining an end effector inside, at, or outside a sphere centered on a target position, the second one is the capability to control directly the rotation of a bone relative to its parent. Those interested in the mathematics can find a short description of the method used to build the Jacobian here.

iTaSC accepts a mix of constraints, and multiple constraints per bone: the solver computes the optimal pose according to the respective weights of each constraint. This is a major improvement from the current constraint system where constraints are solved one by one in order of definition so that conflicting constraints overwrite each other.

Precision

> The maximum variation of the end effector between two successive iterations at which a pose is obtained that is stable enough and the solver should stop the iterations. Lower values means higher precision on the end effector position.

Iterations

> The upper bound for the number of iterations.

Solver

> Selects the inverse Jacobian solver that iTaSC will use.

SDLS

Computes the damping automatically by estimating the level of ‘cancellation’ in the armature kinematics. This method works well with the Copy Pose constraint but has the drawback of damping more than necessary around the singular pose, which means slower movements. Of course, this is only noticeable in Simulation mode.

DLS

Computes the damping manually which can provide more reactivity and more precision.

Damping Max

Maximum amount of damping. Smaller values means less damping, hence more velocity and better precision but also more risk of oscillation at singular pose. 0 means no damping at all.

Damping Epsilon

Range of the damping zone around singular pose. Smaller values means a smaller zone of control and greater risk of passing over the singular pose, which means oscillation.

Note

Damping and *Epsilon* must be tuned for each armature. You should use the smallest values that preserve stability.

Note

- The SDLS solver does not work together with a Distance constraint. You must use the DLS solver if you are going to have a singular pose in your animation with the Distance constraint.
- Both solvers perform well if you do not have a singular pose.

Animation

In Animation mode, iTaSC operates like an IK solver: it is stateless and uses the pose from F-Curves interpolation as the start pose before the IK convergence. The target velocity is ignored and the solver converges until the given precision is obtained. Still the new solver is usually faster than the old one and provides features that are inherent to iTaSC: multiple targets per bone and multiple types of constraints.

Simulation

The Simulation mode is the stateful mode of the solver: it estimates the target's velocity, operates in a 'true time' context, ignores rotation from keyframes (except via a joint rotation constraint) and builds up a state cache automatically.

Reiteration

Never

The solver starts from the rest pose and does not reiterate (converges) even for the first frame. This means that it will take a few frames to get to the target at the start of the animation.

Initial

The solver starts from the rest pose and re-iterates until the given precision is achieved, but only on the first frame (i.e. a frame which doesn't have any previous frame in the cache). This option basically allows you to choose a different start pose than the rest pose and it is the default value. For the subsequent frames, the solver will track the target by integrating the joint velocity computed by the Jacobian solver over the time interval that the frame represents. The precision of the tracking depends on the feedback coefficient, number of substeps and velocity of the target.

Always

The solver re-iterates on each frame until the given precision is achieved. This option omits most of the iTaSC dynamic behavior: the maximum joint velocity and the continuity between frames is not guaranteed anymore in compensation of better precision on the end effector positions. It is an intermediate mode between *Animation* and real-time *Simulation*.

Auto Step

Use this option if you want to let the solver set how many substeps should be executed for each frame. A substep is a subdivision on the time between two frames for which the solver evaluates the IK equation and updates the joint position. More substeps means more processing but better precision on tracking the targets. The auto step algorithm estimates the optimal number of steps to get the best trade-off between processing and precision. It works by estimation of the nonlinearity of the pose and by limiting the amplitude of joint variation during a substep. It can be configured with next two parameters:

Min

Proposed minimum substep duration (in second). The auto step algorithm may reduce the substep further based on joint velocity.

Max

Maximum substep duration (in second). The auto step algorithm will not allow substep longer than this value.

Steps

If Auto Step is disabled, you can choose a fixed number of substeps with this parameter. Substep should not be longer than 10 ms, which means the number of steps is 4 for a 25 fps animation. If the armature seems unstable (vibrates) between frames, you can improve the stability by increasing the number of steps.

Feedback

Coefficient on end effector position error to set corrective joint velocity. The time constant of the error correction is the inverse of this value. However, this parameter has little effect on the dynamic of the armature since the algorithm evaluates the target velocity in any case. Setting this parameter to 0 means ‘opening the loop’: the solver tracks the velocity but not the position; the error will accumulate rapidly. Setting this value too high means an excessive amount of correction and risk of instability. The value should be in the range 20-100. Default value is 20, which means that tracking errors are corrected in a typical time of 100-200 ms. The feedback coefficient is the reason why the armature continues to move slightly in Simulation mode even if the target has stopped moving: the residual error is progressively suppressed frame after frame.

Max Velocity

Indicative maximum joint velocity in radian per second. This parameter has an important effect on the armature dynamic. Smaller value will cause the armature to move slowly and lag behind if the targets are moving rapidly. You can simulate an inertia by setting this parameter to a low value.

Bone IK Panel

Reference

Mode::	Pose Mode
Panel::	*Properties* ‣ *Bone* ‣ *Inverse Kinematics*

This panel is used to control how the *Pose Bones* work in the IK chain.

IK Stretch

Stretch influence to IK target.

Lock

Disallow movement around the axis.

Stiffness

Stiffness around the axis. Influence disabled if using *Lock*.

Limit

Limit movement around the axis.

iTaSC Solver

If the iTaSC IK Solver is used, the bone IK panel changes to add these additional parameters.

Control Rotation

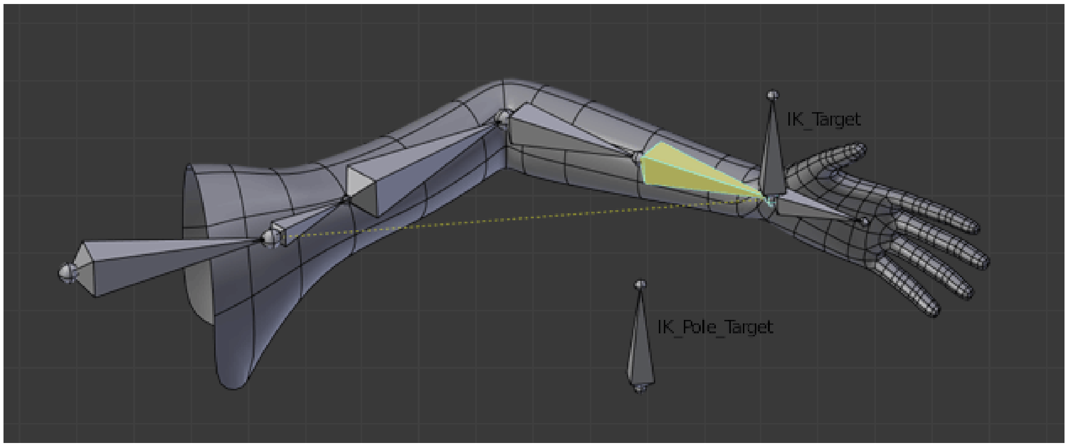
Activates a joint rotation constraint on that bone. The pose rotation computed from Action or UI interaction will be converted into a joint value and passed to the solver as target for the joint. This will give you control over the joint while the solver still tracks the other IK targets. You can use this feature to give a preferred pose for joints (e.g. rest pose) or to animate a joint angle by playing an action on it.

Weight

The importance of the joint rotation constraint based on the constraints weight in case all constraints cannot be achieved at the same time. For example, if you want to enforce strongly the joint rotation, set a high weight on the joint rotation constraint and a low weight on the IK constraints.

Arm Rig Example

This arm uses two bones to overcome the twist problem for the forearm. IK locking is used to stop the forearm from bending, but the forearm can still be twisted manually by pressing `R Y Y` in *Pose Mode*, or by using other constraints.



IK Arm Example.

Note that, if a *Pole Target* is used, IK locking will not work on the root bone.

Spline IK

Spline IK is a constraint which aligns a chain of bones along a curve. By leveraging the ease and flexibility of achieving aesthetically pleasing shapes offered by curves and the predictability and well-integrated control offered by bones, Spline IK is an invaluable tool in the riggers' toolbox. It is particularly well suited for rigging flexible body parts such as tails, tentacles, and spines, as well as inorganic items such as ropes.

Full description of the settings for the spline IK can be found on the Spline IK page.

Basic Setup

The Spline IK Constraint is not strictly an *Inverse Kinematics* method (i.e. IK Constraint), but rather a *Forward Kinematics* method (i.e. normal bone posing). However, it still shares some characteristics of the IK Constraint, such as operating on multiple bones, not being usable for Objects, and being evaluated after all other constraints have been evaluated. It should be noted that if a Standard IK chain and a Spline IK chain both affect a bone at the same time the Standard IK chain takes priority. Such setups are best avoided though, since the results may be difficult to control.

To setup Spline IK, it is necessary to have a chain of connected bones and a curve to constrain these bones to:

1. With the last bone in the chain selected, add a Spline IK Constraint from the Bone Constraints tab in the Properties.
2. Set the *Chain Length* setting to the number of bones in the chain (starting from and including the selected bone) that should be influenced by the curve.
3. Finally, set the *Target* field to the curve that should control the curve.

Congratulations, the bone chain is now controlled by the curve.

Settings and Controls

For the precise list of options, see Spline IK constraint. This section is intended to introduce the workflow.

Roll Control

To control the Roll of the Spline IK chain, the standard methods of rotating the bones in the chain along their local Y axes still apply. For example, start at the farthest bone and simply rotate the bones in the chain around their local Y axes to adjust the roll of the chain from that point onward.

Applying Copy Rotation constraints on the bones also works.

Note

There are a couple of limitations to consider:

- Bones do not inherit a curve's tilt value to control their roll.
- There is no way of automatically creating a twisting effect where a dampened rotation is inherited up the chain. Consider using Bendy Bones instead.

Offset Controls

The entire bone chain can be made to follow the shape of the curve while still being able to be placed at an arbitrary point in 3D space when the *Chain Offset* option is enabled. By default, this option is not enabled, and the bones will be made to follow the curve in its untransformed position.

Length Control

The *Y Scale Mode* setting can be used to choose the way bones are scaled length-wise. The available options allow stretching the bone chain to fit the curve, using the pre-IK scaling, or doing neither. In addition, the scale of the curve Object affects the result.

Thickness Controls

The thickness of the bones in the chain is controlled using the constraint's *XZ Scale Mode* setting. This setting determines the method used for determining the scaling on the X and Z axes of each bone in the chain.

The available modes are:

None
: This option keeps the X and Z scaling factors as 1.0.

Volume Preserve
: The X and Z scaling factors are taken as the inverse of the Y scaling factor (length of the bone), maintaining the 'volume' of the bone.

Bone Original
: This options just uses the X and Z scaling factors the bone would have after being evaluated in the standard way.

In addition to these modes, there is an option, *Use Curve Radius*. When this option is enabled, the average radius of the radii of the points on the curve where the joints of each bone are placed, are used to derive X and Z scaling factors. This allows the scaling effects, determined using the modes above, to be tweaked as necessary for artistic control.

Tips for Nice Setups

- For optimal deformations, it is recommended that the bones are roughly the same length, and that they are not too long, to facilitate a better fit to the curve. Also, bones should ideally be created in a way that follows the shape of the curve in its 'rest pose' shape, to minimize the problems in areas where the curve has sharp bends which may be especially noticeable when stretching is disabled.

- For control of the curve, it is recommended that hooks (in particular, Bone Hooks) are used to control the control points of the curve, with one hook per control point. In general, only a few control points should be needed for the curve (e.g. one for every 3-5 bones offers decent control).
- The type of curve used does not really matter, as long as a path can be extracted from it that could also be used by the Follow Path Constraint. This really depends on the level of control required from the hooks.

Lattice

Lattice – or commonly called deformation cage outside of Blender. A lattice consists of a three-dimensional non-renderable grid of vertices. Its main use is to apply a deformation to the object it controls with a Lattice Modifier. If the object is parented with *Lattice Deform* a Lattice Modifier is automatically applied.

Editing

Flip (Distortion Free)

Mirrors the vertices displacement from their base position.

U, V, W

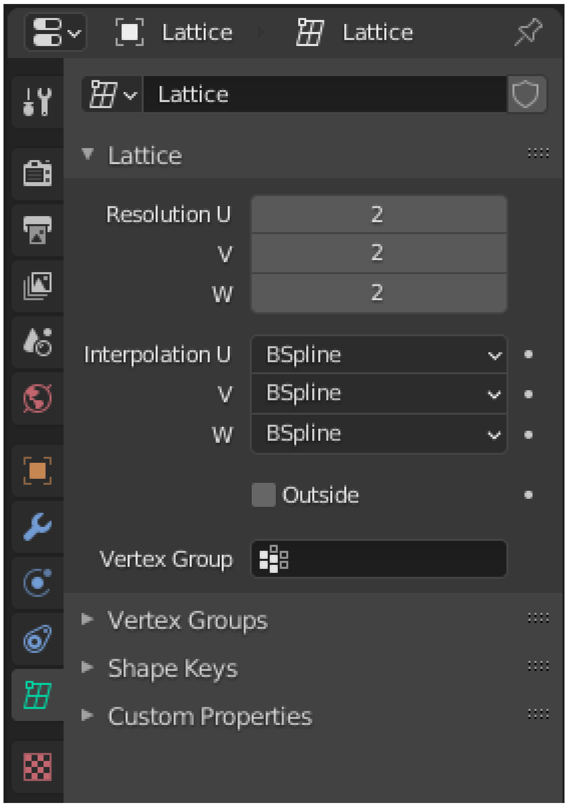
Make Regular

Resets the whole lattice to a regular grid, where the cells are scaled to one cubic unit.

Properties

Lattice

A Data-Block Menu.


Lattice
Lattice
Lattice
Lattice
Vertex Groups
Shape Keys
Custom Properties

Resolution U 2

V 2

W 2

Interpolation U BSpline

V BSpline

W BSpline

☐ Outside

Vertex Group

Lattice properties.

Lattice

Points

Rate of subdivision in the axes:

U, V, W

Interpolation Type

Selector for each axis. See [Different types of interpolation.](#).

Linear, Cardinal, Catmull-Rom, B-Spline

Outside

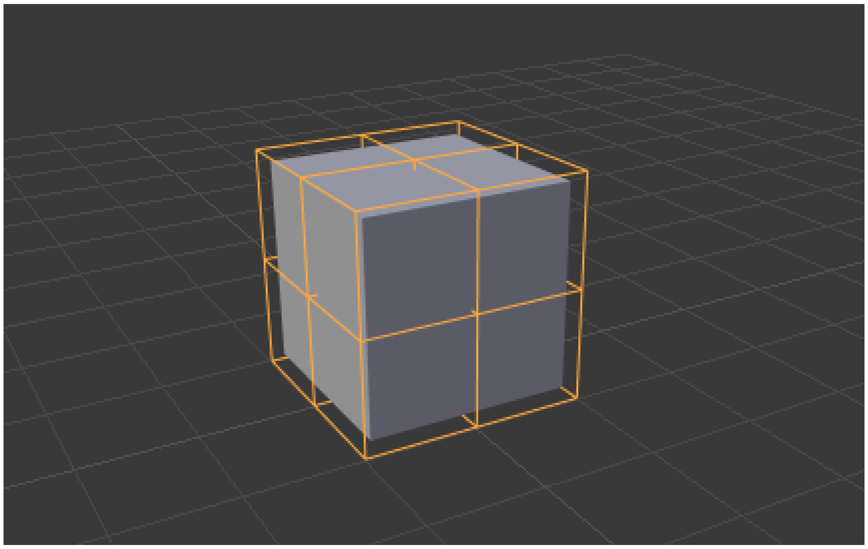
Takes only the vertices on the surface of the lattice into account.

Vertex Group

The strength of the influence assigned as a weight to the individual vertices in the selected vertex group.

Usage

The lattice should be scaled and moved to fit around your object in Object Mode. Any scaling applied to the object in Edit Mode will result in the object deforming. This includes applying its scale with `Ctrl-A` as this will achieve the same result as scaling the lattice in Edit Mode, and therefore the object.



Lattice around the cube object in Object Mode.

Constraints

- Introduction
    - Adding & Removing Constraints
    - Tips
- Interface
    - Header
    - Common
    - Stack

Constraint Types

- Motion Tracking
    - Camera Solver Constraint
    - Object Solver Constraint
    - Follow Track Constraint
- Transform
    - Copy Location Constraint
    - Copy Rotation Constraint
    - Copy Scale Constraint
    - Copy Transforms Constraint
    - Limit Distance Constraint
    - Limit Location Constraint
    - Limit Rotation Constraint
    - Limit Scale Constraint
    - Maintain Volume Constraint
    - Transformation Constraint
    - Transform Cache Constraint
- Tracking
    - Clamp To Constraint
    - Damped Track Constraint
    - Inverse Kinematics Constraint
    - Locked Track Constraint
    - Spline IK Constraint

    - Stretch To Constraint
    - Track To Constraint
- Relationship
    - Action Constraint
    - Armature Constraint
    - Child Of Constraint
    - Floor Constraint
    - Follow Path Constraint
    - Pivot Constraint
    - Shrinkwrap Constraint

Introduction

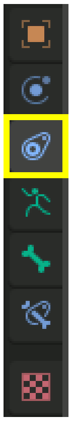
Constraints are a way to control an object’s properties (e.g. its location, rotation, scale), using either plain static values (like the “limit” ones), or another object, called “target” (like e.g. the “copy” ones).

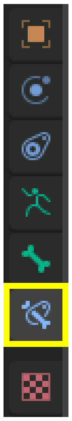
Even though constraints are useful in static projects, their main usage is obviously in animation.

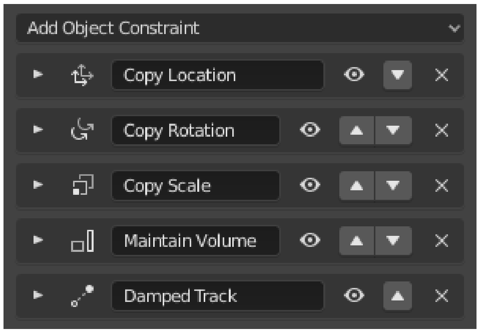
- You can control an object’s animation through the targets used by its constraints (this is a form of indirect animation). Indeed, these targets can then control the constraint’s owner’s properties, and hence, animating the targets will indirectly animate the owner.
- You can animate constraints’ settings. e.g. the *Influence* or when using an armature’s bone as target, animate where along this bone (between root and tip) lays the real target point.

They can make the eyes of a tennis player track a tennis ball bouncing across the court, allow the wheels on a bus to all rotate together, help a dinosaur’s legs bend at the knee automatically, and make it easy for a hand to grip the hilt of a sword and the sword to swing with the hand.

Constraints, in Blender, work with Objects and Bones. Read about using constraints in rigging in the Armature chapter.







Object	Bone	The Constraint Stack is evaluated from top to bottom.

Constraints work in combination with each other to form a Constraint Stack.

Adding & Removing Constraints

To add a constraint click on the *Add Object Constraint* menu in the Constraints tab. Alternatively, you can use the Add Constraint (with Targets) operator.

To copy constraints from one object to another use Copy Constraints to Selected Objects.

Any single constraint can be removed by clicking on the “X” button in the constraint’s header. To remove all constants from an object use Clear Object Constraints.

Tip

Tracking constraints can be added/removed using the Track menu.

Tips

Constraints are a fantastic way to add sophistication and complexity to a rig.

But be careful not to rush in too quickly, piling up constraint upon constraint until you lose all sense of how they interact with each other.

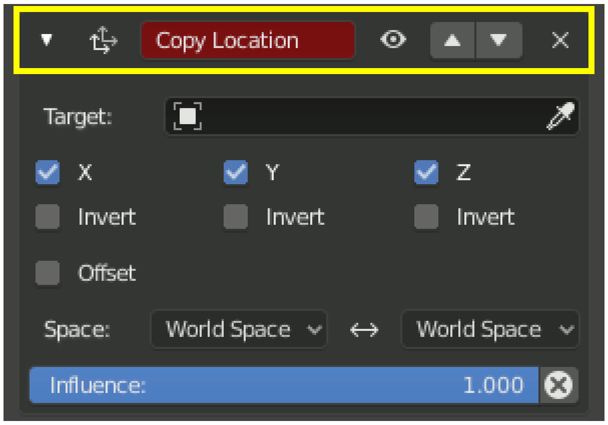
Start simply. Get to know a single constraint inside and out. Copy Location Constraint is a good first constraint to explore it also has an animation example. Take the time to understand every fundamental concept behind it, and the other constraints will make far more sense.

Interface

- Header
- Common
    - Target
    - Space
    - Influence
- Stack

Header

Every constraint has a header. The interface elements of the header are explained below using a Copy Location constraint as an example.



Copy Location

Target:

☑ X ☑ Y ☑ Z

☐ Invert ☐ Invert ☐ Invert

☐ Offset

Space: World Space ↔ World Space

Influence: 1.000

A Header sits at the top of every constraint.

Expand (down/right arrow icon)

Show or Hide the settings of the constraint. Tidy up the constraint stack by hiding constraints that do not currently need attention. Constraints will continue to affect the scene even when hidden.

Icon

The constraint type icon.

Name

Give the constraint a meaningful name in this text field, which describes its purpose. Meaningful names help you and your team members understand what each constraint is supposed to do.

The *red* background is a warning that the constraint is not yet functional. The background will turn *gray* when the constraint is functioning. When this Copy Location constraint has a valid target in the *target field* it will turn gray and begin to function.

Mute (eye icon)

Enable or Disable the constraint. Disabling a constraint will stop its affect on the scene.

Disabling a constraint is useful for turning off a constraint without losing all of its settings. Disabling means you can enable the constraint at a later time with the settings intact. Disabling is similar to setting the Influence to 0.0.

Extras

Apply `Ctrl-A`

Makes the constraint “real” by applying any transformations caused by the constraint to make the original object to match the results of the constraint and deletes the constraint.

Warning

Applying a constraint that is not first in the stack will ignore the stack order (it will be applied as if it was the first one), and may produce undesired results.

Duplicate `Shift-D`

Creates a duplicate of the constraint just below current one in the stack.

Copy to Selected

Copies the constraint from the Active object to all selected objects.

Move to First/Last

Moves the constraint to the first or last position in the constraint stack.

(Delete) `X`, `Delete`

Delete the constraint from the stack. The settings will be lost. The constraint will no longer affect the final outcome of the stack.

Move `::::`

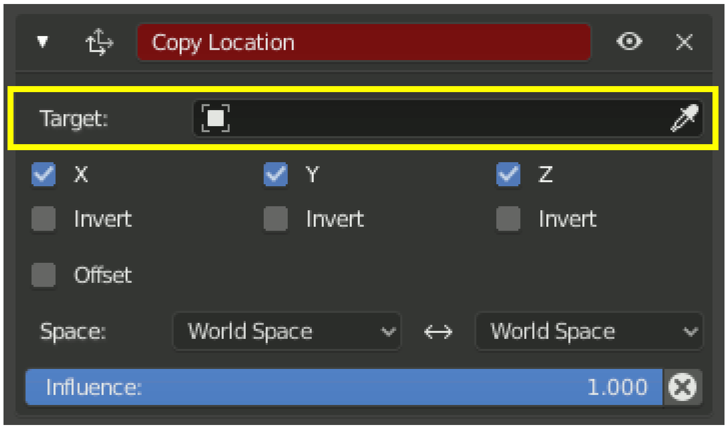
Move a constraint up or down in the constraint stack. Since the stack is evaluated from top to bottom, moving a constraint in the stack can significantly affect the final outcome of the stack.

- If there is only one constraint in the stack, the arrows will not be displayed.
- If the constraint is at the top of the stack, only the down arrow will be displayed.
- If the constraint is at the bottom of the stack, only the up arrow will be displayed.

Common

Target

The Target Data ID field lets you link the constraint to a Target object of your choosing. This link provides data to the constraint so that it can begin to function. For example, the Copy Location Constraint needs location data to function. Fill in the Target field, and the Copy Location constraint will begin to use location data from the Target object.



Target:

☑ X ☑ Y ☑ Z

☐ Invert ☐ Invert ☐ Invert

☐ Offset

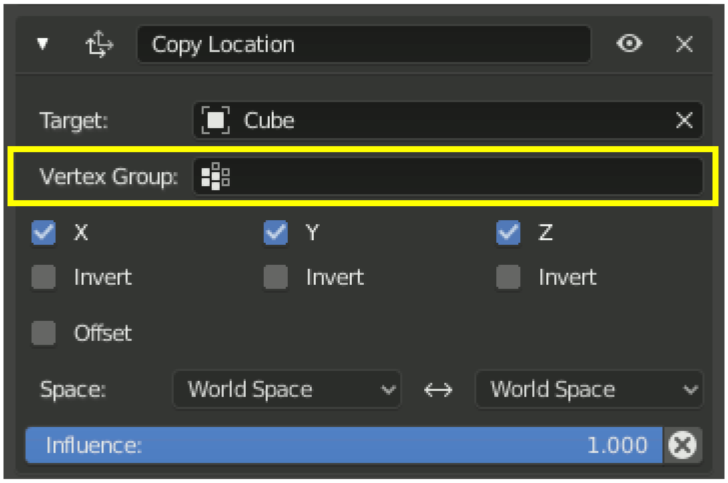
Space: World Space ↔ World Space

Influence: 1.000

The Target field must be filled in for the constraint to function.

By default, the Target will use the Object Origin as the target point.

If the Target field links to a Mesh or Lattice object, a Vertex Group field will appear. Enter the name of a vertex group and the constraint will target the median point of this vertex group instead of the object's origin.



Copy Location

Target: Cube

Vertex Group:

☑ X ☑ Y ☑ Z

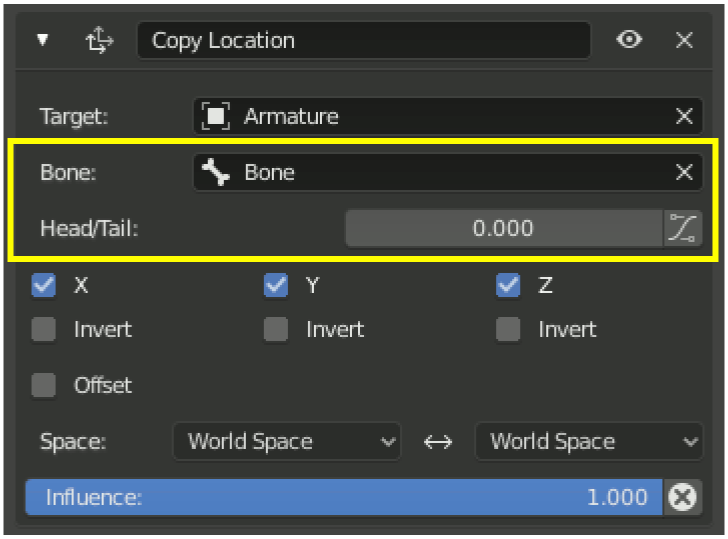
☐ Invert ☐ Invert ☐ Invert

☐ Offset

Space: World Space ↔ World Space

Influence: 1.000

If the Target field links to an Armature, a Bone field will appear along with a *Head/Tail* slider. Enter the name of a bone and the constraint will target the bone instead of the entire armature object origin.



Copy Location

Target: Armature

Bone: Bone

Head/Tail: 0.000

☑ X ☑ Y ☑ Z

☐ Invert ☐ Invert ☐ Invert

☐ Offset

Space: World Space ↔ World Space

Influence: 1.000

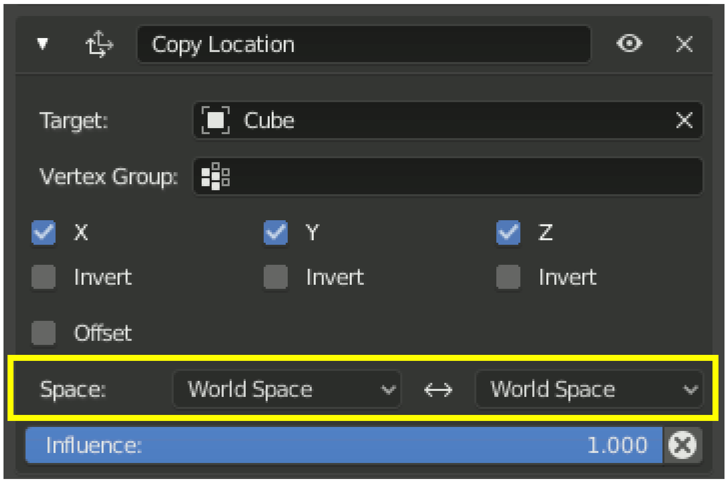
The slider moves the precise position of the target between the Head and Tail of the bone. Some constraints have a button next to the slider that enables using the curved shape of Bendy Bones.

Space

Constraints need a frame of reference in order to function. This frame of reference is called the “space” of the constraint. Choosing one space vs.

another will change this frame of reference and substantially alter the behavior of a constraint.

To understand how changing the space will change the behavior of the constraint, consider experimenting with two empties. Make sure they display as arrows so that you can see the local axes for each empty. Make sure to size one empty a little larger than the other so that they are both always visible even if directly on top of each other. Then add a constraint to one empty that targets the other and experiment thoroughly by moving, rotating and scaling the target in many different ways.


Copy Location

Target: Cube

Vertex Group:

☑ X ☑ Y ☑ Z

☐ Invert ☐ Invert ☐ Invert

☐ Offset

Space: World Space ↔ World Space

Influence: 1.000

This constraint is set to use World Space as the frame of reference for both its *Target* space and its *Owner* space.

Target Space & Owner Space

The space used to evaluate the target of the constraint is called the *Target* space. The space used to evaluate the constrained object (the object that owns the constraint) is called the *Owner* space. Hover over the space select menu(s) to learn whether it affects the space of the target or the space of the owner.

When the constraints use a *Target* and/or/nor an *Owner* space there will be no, one or two selector(s). The Copy Location constraint in example use both Target **and** *Owner* space.

When a constraint uses both *Target* and *Owner* space, the Target and Owner can be any combination of space types.

Space Types

World Space

> In this space type the world is the frame of reference for the object (or bone). Location is relative to the world origin. Rotation and Scale are oriented to the world axes. Transformations to the object, the object's parent and any other constraints higher up in the constraint stack are all taken into account.

Local Space

> This space excludes all effects of the parent objects or bones, as well as the rest position and orientation of the bone itself. Only transformations applied to the object or bone itself are taken into account.
>
> **Warning**
>
> For objects without a parent Local Space has a special meaning, different from the normal behavior of local space for bones or objects that have a parent. This behavior is kept for backwards compatibility, but may be removed in the future and shouldn't be used.

Local with Parent *Bones Only*

> The bone position and orientation is evaluated relative to its rest pose location and orientation, thus including both its own transformations and those caused by a possible parent relationship (i.e. the chain's transformations above the bone).

Pose Space *Bones Only*

> The bone position and orientation is evaluated in the armature object local space (i.e. independently from the armature transformations in *Object Mode*). Hence, if the armature object has null transformations, *Pose Space* will have the same effect as *World Space*.

Custom Space

The position and orientation is evaluated relative to the current position and orientation of an arbitrary object or bone that is specified via additional input fields that appear when this option is selected. This can be used to evaluate the constraint using an arbitrary coordinate system.

Local Space (Owner Orientation) *Bone Targets Only*

This space works like *Local Space*, with an additional coordinate space transformation that compensates for the difference in the rest pose orientations of the owner and target bones. If applied as the *Local Space* of the owner, this will produce the same global space movement as the target, provided parents are still at rest pose.

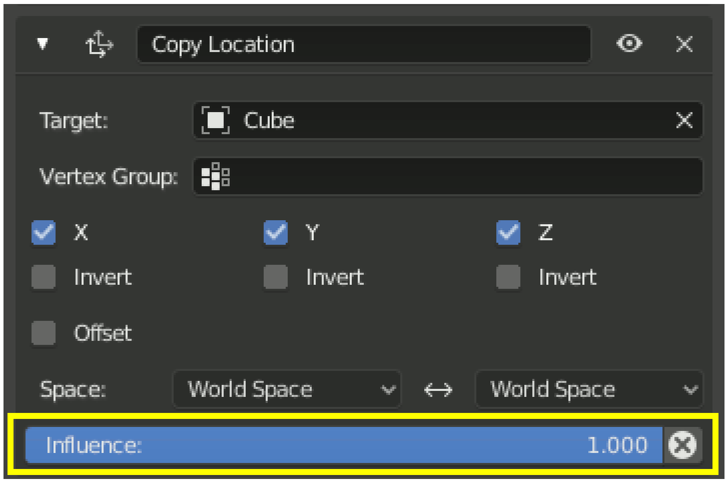
This option replaces the following setup with two additional bones:

1. An extra child bone of the target, rotated the same as the owner in rest pose.
2. An extra sibling bone of the target, positioned same as the child in rest pose and using Copy Transforms in *World Space* from the child.
3. The constraint uses *Local Space* of the sibling instead of the original target.

This video demonstrates the difference from ordinary *Local Space*:

Influence

The influence slider determines how much the constraint will affect the constrained object (target).



Copy Location

Target: Cube

Vertex Group:

☑ X ☑ Y ☑ Z

☐ Invert ☐ Invert ☐ Invert

☐ Offset

Space: World Space ↔ World Space

Influence: 1.000

An influence of 0.0 will have no effect. An influence of 1.0 will have the full effect.

Values between (0.0 and 1.0) will have a partial effect, but be careful. These partial effects can be difficult to control, especially as the constraint stack grows in complexity.

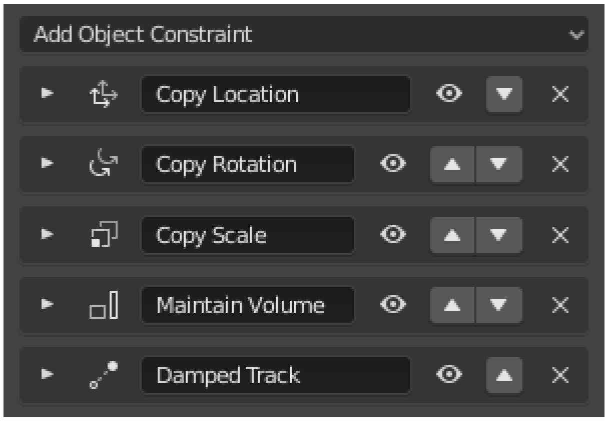
The influence value is animatable, allowing constraints to be turned off, or partially on as needed.

(Disable and Keep Transform)
: Disables the constraint while trying to preserve the current object position. This may not work perfectly if other constraints remain active.

Stack

The combination of all the constraints affecting an object is called the Constraints Stack. The Stack is in the Constraints panel, below the Add Constraint menu.

Constraints in the stack are evaluated from top to bottom. The order of each constraint has a substantial impact on the final outcome of the stack. Changing the order of the constraints can change the behavior of the entire stack.


Add Object Constraint
Copy Location
Copy Rotation
Copy Scale
Maintain Volume
Damped Track

The constraints in this example stack are evaluated from top to bottom starting with the “Copy Location” constraint and ending with the final “Damped Track” constraint.

To change the order of a constraint use the up/down arrows in the header.

Motion Tracking Constraints

- Camera Solver Constraint
  - Options
- Object Solver Constraint
  - Options
- Follow Track Constraint
  - Options
  - Example

Camera Solver Constraint

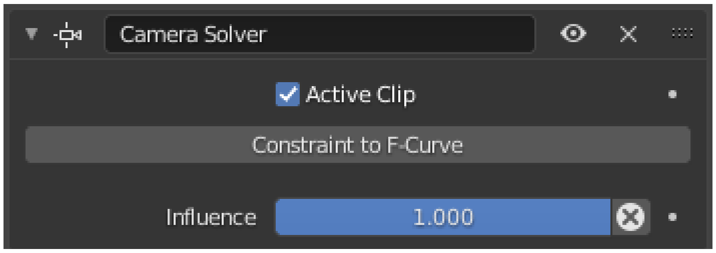
The *Camera Solver* constraint gives the owner of this constraint, the location and rotation of the "solved camera motion".

The "solved camera motion" is where Blender reconstructs the position of the physical, real-world camera, when it filmed the video footage, relative to the thing being tracked.

Note

This constraint only works after you have set up a minimum of eight markers and pressed Solve Camera Motion (*Movie Clip Editor* ‣ *Toolbar* ‣ *Solve* ‣ *Solve Camera Motion*).

Options



Camera Solver Constraint panel.

Active Clip
: Receive tracking data from the scene's Active Clip. If unchecked, an option appears to choose from the other clips.

Constraint to F-Curve
: Applies the constraint, creating Keyframes for the transforms.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Object Solver Constraint

The *Object Solver* constraint gives the owner of this constraint, the location and rotation of the “solved object motion”.

The “solved object motion” is where Blender thinks the physical, real-world (tracked) object was, relative to the camera that filmed it.

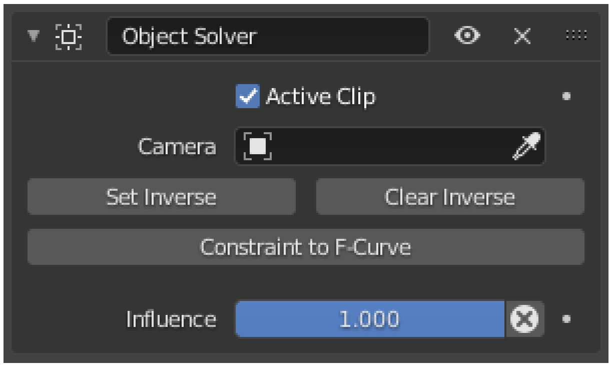
Can be used to add a mesh to video for example.

Note

This constraint only works after you have set up a minimum of eight markers and pressed Solve object Motion. Located at *Movie Clip Editor* ‣ *Toolbar* ‣ *Solve* ‣ *Solve Camera Motion*.

If it says *Solve Camera Motion* instead of *Solve Object Motion* then go into the *Movie Clip Editor* ‣ *Sidebar region* ‣ *Objects* and switch it from the camera, to an object.

Options


Object Solver
Active Clip
Camera
Set Inverse
Clear Inverse
Constraint to F-Curve
Influence
1.000

Object Solver Constraint panel.

Active Clip
Receive tracking data from the scene's Active Clip. If unchecked, an option appears to choose from the other clips.

Object
Select a tracked object to receive transform data from.

Camera
Select the camera to which the motion is parented to (if left empty the active scene camera is used).

Set Inverse
Moves the origin of the object to the origin of the camera.

Clear Inverse
Moves the origin of the object back to the spot set in the Movie Clip Editor *Toolbar* ▸ *Solve* ▸ *Orientation* ▸ *Set Origin*.

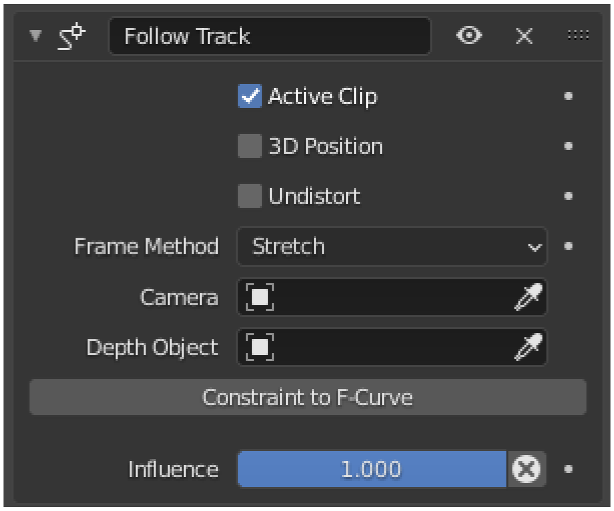
Constraint to F-Curve
Applies the constraint, creating keyframes for the transforms.

Influence
Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Follow Track Constraint

By default the Follow Track constraint is making objects have the same position at a frame as the track has. The motion of this object happens on a single plane defined by the camera and the original position of the object.

Options



☑ Active Clip

☐ 3D Position

☐ Undistort

Frame Method: Stretch

Camera

Depth Object

Influence: 1.000

Follow Track Constraint panel.

Active Clip
: Receive tracking data from the scene's [Active Clip](). If unchecked, an option appears to choose from the other clips.

3D Position
: Use the 3D position of the track to parent to.

Undistorted
: Parent to the undistorted position of the 2D track.

Frame Method
: Defines how the footage is fitted in the camera frame.

Camera

Select the camera to which the motion is parented to (if empty, the active scene camera is used).

Depth Object

If this object is set, constrained objects will be projected onto the surface of this depth object which can be used to create facial makeup visual effects.

Constraint to F-Curve

Creates F-Curves for the object that copies the movement caused by the constraint.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Follow Track Example Video

Transform Constraints

- Copy Location Constraint
- Copy Rotation Constraint
- Copy Scale Constraint
- Copy Transforms Constraint
- Limit Distance Constraint
- Limit Location Constraint
- Limit Rotation Constraint
- Limit Scale Constraint
- Maintain Volume Constraint
- Transformation Constraint
- Transform Cache Constraint

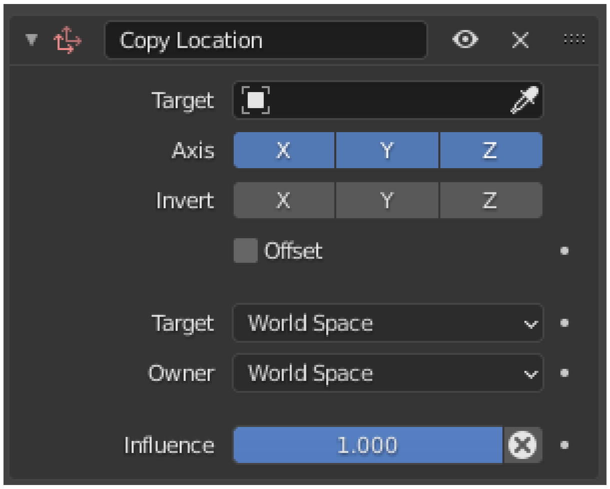
Copy Location Constraint

The *Copy Location* constraint forces its owner to have the same location as its target.

Important

Note that if you use such a constraint on a *connected* bone, it will have no effect, as it is the parent's tip which controls the position of your owner bone's root.

Options



Target: World Space
Owner: World Space
Influence: 1.000

Copy Location panel.

Target
: [Data ID](#) used to select the constraints target, and is not functional (red state) when it has none. See [common constraint properties](#) for more information.

Axis
: These buttons control which axes are constrained.

Invert

Invert their respective corresponding axis coordinates.

Offset

When enabled, this control allows the owner to be moved (using its current transform properties), relative to its target’s position.

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Examples

Animation

Let us animate a solar system with the *Copy Location* constraint and its *Offset* option. You can make the owner, called “moon”, describe perfect circles centered on the world origin (using e.g. *Location X/Y* sine and cosine F-Curves, see Built-in Function Modifier). Then copy the location of a target “earth” with the *Offset* checkbox enabled to model a satellite in a (simplified) orbit around its planet. Repeat these steps for more planets circling around its center star “sun”.

Following video is a small animation of a solar system created using (among a few others) the previously described technique:

Note that, this ‘solar’ system is not realistic at all (the wrong scale, the earth is rotating in the wrong direction around the sun, …).

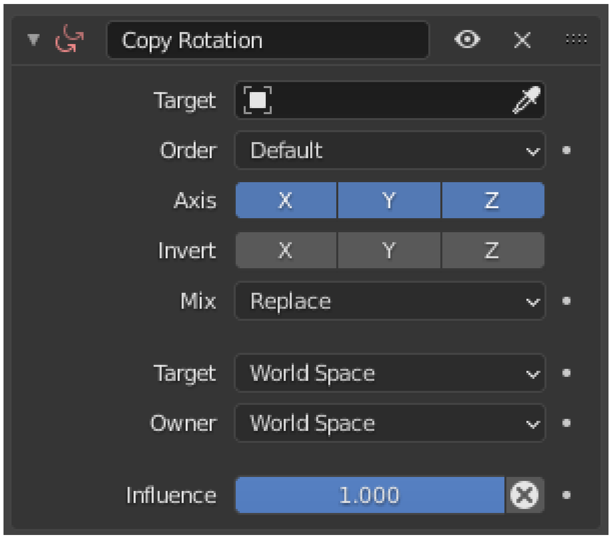
You can download the blend-file used to create this animation.

Furthermore you can also animate a few properties of each constraint using animation curves: e.g. you can animate the *Influence* of a constraint. It is used to first let the camera follow the moon, then the earth, and finally using two *Copy Location* constraints with *Offset* set.

Copy Rotation Constraint

The *Copy Rotation* constraint forces its owner to match the rotation of its target.

Options


Copy Rotation
Target
Order: Default
Axis: X Y Z
Invert: X Y Z
Mix: Replace
Target: World Space
Owner: World Space
Influence: 1.000

Copy Rotation panel.

Target
: Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Order
: Allows specifying which Euler order to use during the copy operation. Defaults to the order of the owner.

Axis
: These buttons control which axes are constrained.

Invert

Invert their respective corresponding axis coordinates.

Mix

Specifies how the new rotation is combined with the existing rotation.

Replace

The new axis values replace existing values.

Add

The new axis values are added to the existing values.

Before Original

The new rotation is added before the existing rotation, as if it was applied to a parent of the constraint owner.

After Original

The new rotation is added after the existing rotation, as if it was applied to a child of the constraint owner.

Offset (Legacy)

This replicates the behavior of the original Offset checkbox. It was intended to be similar to the *Before Original* behavior, but does not work correctly with multiple axis rotations, and is thus deprecated.

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

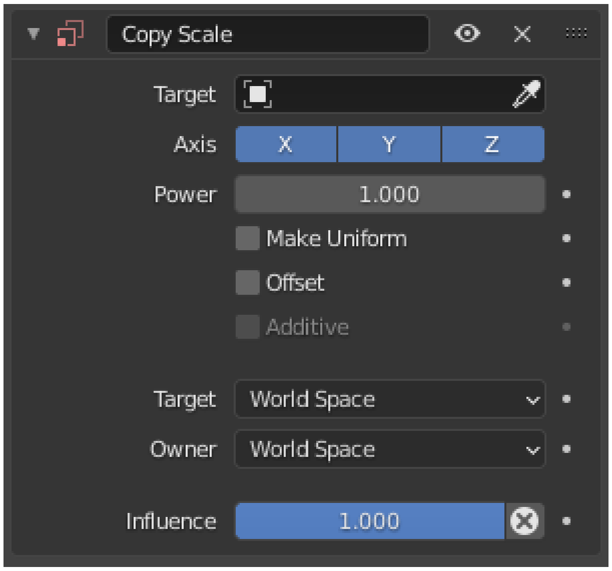
Copy Scale Constraint

The *Copy Scale* constraint forces its owner to have the same scale as its target.

Note

Here we talk of *scale*, not of *size*! Indeed, you can have two objects, one much bigger than the other, and yet both of them have the same scale. This is also true with bones: in *Pose Mode*, they all have a unitary scale when they are in rest position, represented by their visible length.

Options



Copy Scale

Target

Axis: X Y Z

Power: 1.000

☐ Make Uniform

☐ Offset

☐ Additive

Target: World Space

Owner: World Space

Influence: 1.000

Copy Scale panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Axis
: These buttons control which axes of the target scale are copied.

Power
: Allows raising the copied scale to the specified arbitrary power.

Make Uniform
: Instead of copying scale for individual axes, apply a uniform scaling factor to all axes of the owner that achieves the same overall change in volume.

Offset
: When enabled, the constraint combines the copied scale with the owner's scale, instead of overwriting it.

Additive
: Uses addition instead of multiplication in the implementation of the *Offset* option.

Target/Owner
: Standard conversion between spaces. See common constraint properties for more information.

Influence
: Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Note

Since scale is a multiplicative quantity, it should be combined using multiplication, and split into fractions or inverted via power. Thus the use of *Power* is more mathematically correct than *Influence*, which uses linear interpolation. The use of the *Additive* option is also not recommended.

Tip

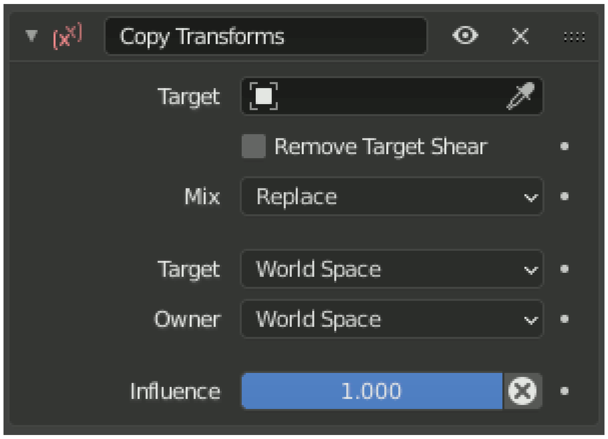
To copy scale from one axis of the target to all axes of the owner, disable other axes, enable *Make Uniform*, and set *Power* to 3.

Example

Copy Transforms Constraint

The *Copy Transforms* constraint forces its owner to have the same transforms as its target.

Options


Copy Transforms
Target
Remove Target Shear
Mix
Replace
Target
World Space
Owner
World Space
Influence
1.000

Copy Transforms panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Remove Target Shear

Removes shearing from the target transformation after the target space conversion, ensuring it consists purely of translation, rotation and scale. Note that Copy Rotation always does this.

Mix

Specifies how the copied transformation is combined with the existing transformation.

Replace

The new transformation replaces the existing transformation.

Before/After Original (Full)

The new transformation is added before/after the existing transformation, as if it was applied to an imaginary parent/child of the constraint owner. Scale is handled like in the most basic Full Inherit Scale mode of bones, so combining non-uniform scale and rotation will create shear.

Before/After Original (Aligned)

The new transformation is added before/after the existing transformation, as if it was applied to an imaginary parent/child of the constraint owner. Scale is handled like in the Aligned Inherit Scale mode of bones to avoid creating shear.

This is equivalent to using the *Split Channels* option, but replacing the location component with the result of *Full*. If only uniform scale is used, the result is identical to *Full*.

Before/After Original (Split Channels)

Combines location, rotation and scale components of the transformation separately, similar to a sequence of three Copy Location, Copy Rotation and Copy Scale (with Offset) constraints bundled together in one operation; the result may be slightly different in case of sheared inputs.

Unlike *Aligned*, in this mode location channels are simply added together, so rotation and scale components of the input transformations cannot affect the resulting location.

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

This video shows the difference between the mix modes. The right input is mixed after the left one:

A general demonstration of the constraint:

Limit Distance Constraint

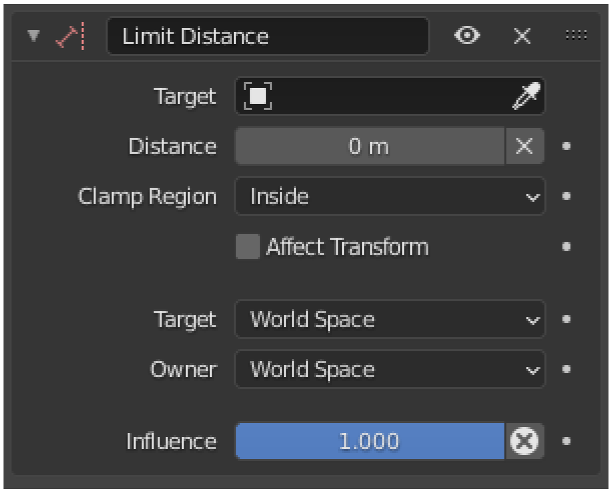
The *Limit Distance* constraint forces its owner to stay either further from, nearer to, or exactly at a given distance from its target. In other words, the owner's location is constrained either outside, inside, or at the surface of a sphere centered on its target.

When you specify a (new) target, the *Distance* value is automatically set to correspond to the distance between the owner and this target.

Important

Note that if you use such a constraint on a *connected* bone, it will have no effect, as it is the parent's tip which controls the position of your owner bone's root.

Options


Limit Distance
Target
Distance: 0 m
Clamp Region: Inside
☐ Affect Transform
Target: World Space
Owner: World Space
Influence: 1.000

Limit Distance panel.

Target

Data ID used to select the constraint's target, and is not functional (red state) when it has none. See common constraint properties for more information.

Distance

This number field sets the limit distance, i.e. the radius of the constraining sphere.

(Reset Distance)

Resets the *Distance* value,so that it corresponds to the actual distance between the owner and its target (i.e. the distance before this constraint is applied).

Clamp Region

The *Limit Mode* select menu allows you to choose how to use the sphere defined by the *Distance* setting and target's origin:

Inside

The owner is constrained *inside* the sphere.

Outside

The owner is constrained *outside* the sphere.

Surface

The owner is constrained *on the surface* of the sphere.

Affect Transform

Transform operators will take the constraint into account to immediately restrict the resulting transform property values.

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Tip

Evaluating both owner and target in a Custom Space using the root bone or any other suitable parent bone will automatically scale the effective distance with the relevant part of the rig.

Example

Limit Location Constraint

An object or *unconnected* bone can be moved around the scene along the X, Y and Z axes. This constraint restricts the amount of allowed translations along each axis, through lower and upper bounds.

The limits for an object are calculated from its origin, and the limits of a bone, from its root.

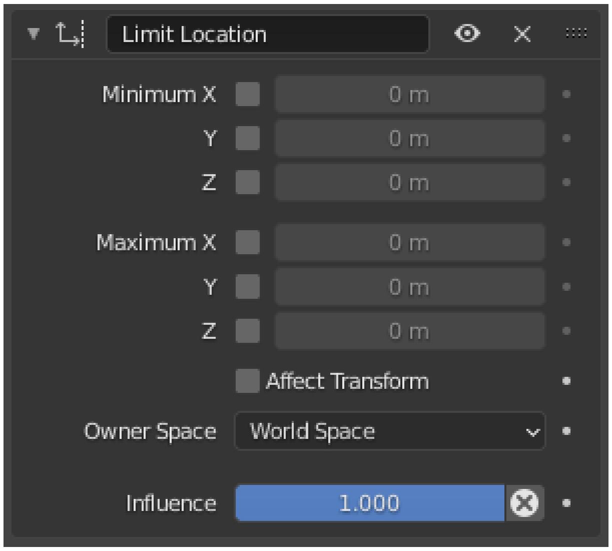
It is interesting to note that even though the constraint limits the visual and rendered location of its owner, its owner's data-block still allows (by default) the object or bone to have coordinates outside the minimum and maximum ranges. This can be seen in its *Transform* panel.

When an owner is selected and attempted to be moved outside the limit boundaries, it will be constrained to those boundaries visually and when rendered, but internally, its coordinates will still be changed beyond the limits. If the constraint is removed, its ex-owner will seem to jump to its internally specified location.

Similarly, if its owner has an internal location that is beyond the limits, dragging it back into the limit area will appear to do nothing until the internal coordinates are back within the limit threshold (unless you enabled the *Affect Transform* option, see below).

Setting equal the min and max values of an axis, locks the owner's movement along that axis… Although this is possible, using the *Transformation Properties* axis locking feature is probably easier!

Options



Limit Location

Minimum X ☐ 0 m
Y ☐ 0 m
Z ☐ 0 m

Maximum X ☐ 0 m
Y ☐ 0 m
Z ☐ 0 m

☐ Affect Transform

Owner Space: World Space

Influence: 1.000

Limit Location panel.

Minimum X, Y, Z

These buttons enable the lower boundary for the location of the owner’s origin along, respectively, the X, Y and Z axes of the chosen *Space*. The number field below them controls the value of their limit. Note that if a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

Maximum X, Y, Z

These buttons enable the upper boundary for the location of the owner’s origin along, respectively, the X, Y and Z axes of the chosen *Space*. Same options as above.

Affect Transform

As pointed out before by default, even though visually constrained, the owner can still have coordinates out of bounds (as shown by the *Transform* panel). Well, when you enable this checkbox, this is no longer possible – the owner’s transform properties are also limited by the constraint. However, note that, the constraint does not directly modify the coordinates: you have to select its owner one way or another for this to take effect…

Owner

This constraint allows you to choose in which space to evaluate its owner's transform properties. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Limit Rotation Constraint

This constraint restricts the rotation of an object or bone to be inside specified angular limits. The limits are given as Euler rotation ranges (a min and max angle), and a separate range can be given for each of the three axes.

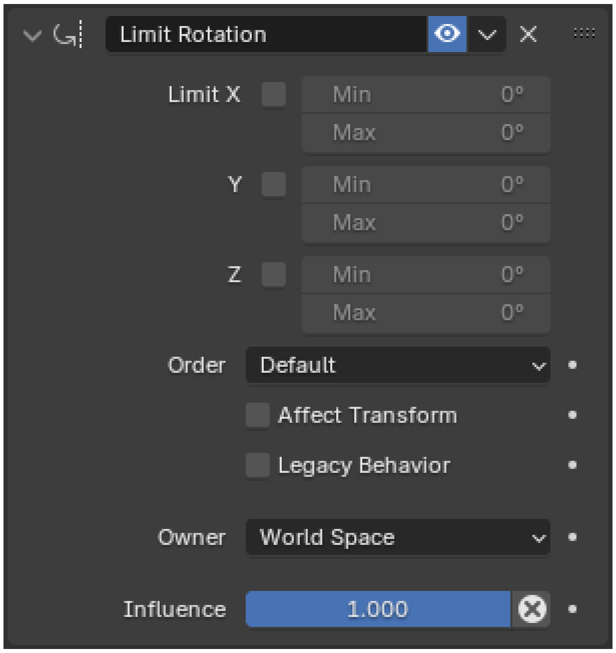
As with all constraints in Blender, this does not (by default) restrict the user-set rotation values of the object/bone as seen in the *Transform* panel. When the object/bone is rotated outside the limit range, it will be constrained to that range in its final displayed/rendered position, but the user-set rotation values will still be outside that range. If the constraint is removed, the object/bone will then jump back to match those user-set values.

Something unique about the Limit Rotation constraint (as compared to the Limit Location and Limit Scale constraints) is that rotations loop, and therefore the meaning of the limit range is subtly different. All constraints in Blender internally work on transform matrices, which can't distinguish between e.g. 180 and -180 degrees, or 0, 360, and 720 degrees. In other words, any angles that result in the same visual rotation are indistinguishable to the constraint system.

What this means for the Limit Rotation constraint is that when the user-set rotation is outside of the limit range, the final displayed rotation will snap to the closest *visual* rotation in that range, not the closest numerical angle. For example, if you have a limit range of 0 to 90 degrees then a user-set rotation of 340 degrees will actually snap to *0 degrees* because that is the closer *visual* rotation, even though 340 is numerically closer to 90.

Note that this constraint does not constrain the bone if it is manipulated by the IK solver. For constraining the rotation of a bone for IK purposes, see Inverse Kinematics.

Options


Limit Rotation
Limit X
Min 0°
Max 0°
Y
Min 0°
Max 0°
Z
Min 0°
Max 0°
Order
Default
Affect Transform
Legacy Behavior
Owner
World Space
Influence
1.000

Limit Rotation panel.

Limit X, Y, Z

These buttons enable the rotation limit around respectively the X, Y and Z axes of the owner, in the chosen *Owner* space. The *Min* and *Max* number fields to their right control the value of their lower and upper boundaries, respectively.

Note

- If a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.
- Unlike the Limit Location constraint, you cannot separately enable lower or upper limits.
- The constraint can be used to simply remove shear from the owner transformation by leaving all limits disabled.

Order

Allows specifying which Euler order to use when applying the limits. Defaults to the order of the owner, or XYZ if the owner uses non-Euler rotations.

Affect Transform

The constraint is taken into account when the object is manually rotated using transformation tools in the editors. This prevents assigning transformation property values (as shown in the *Transform* panel) that exceed the specified limits.

Legacy Behavior

For backwards compatibility: make the constraint behave in the semi-broken way it did prior to Blender 4.2. This old behavior does not properly account for the looping nature of rotations, and therefore causes unpredictable/erratic rotation snapping. However, this behavior can still be useful in some specific circumstances when *Owner* is set to local space, and some older rig setups utilize that. However, that behavior is better and more robustly accomplished with drivers directly on the object/bone's rotation properties, so new rigs should favor that approach over using this option.

Owner

This constraint allows you to choose in which space evaluate its owner's transform properties. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Limit Scale Constraint

An object or bone can be scaled along the X, Y and Z axes. This constraint restricts the amount of allowed scaling along each axis, through lower and upper bounds.

Important

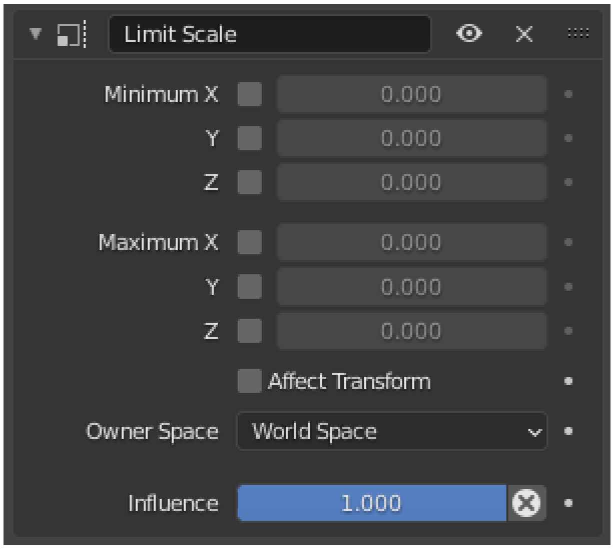
This constraint does not tolerate negative scale values (those you might use to mirror an object…): when you add it to an object or bone, even if no axis limit is enabled, nor the *Affect Transform* option, as soon as you scale your object, all negative scale values are instantaneously inverted to positive ones… And the boundary settings can only take strictly positive values.

It is interesting to note that even though the constraint limits the visual and rendered scale of its owner, its owner's data-block still allows (by default) the object or bone to have scale values outside the minimum and maximum ranges (as long as they remain positive!). This can be seen in its *Transform* panel. When an owner is scaled and attempted to be moved outside the limit boundaries, it will be constrained to those boundaries visually and when rendered, but internally, its coordinates will still be changed beyond the limits. If the constraint is removed, its ex-owner will seem to jump to its internally-specified scale.

Similarly, if its owner has an internal scale that is beyond the limits, scaling it back into the limit area will appear to do nothing until the internal scale values are back within the limit threshold (unless you enabled the *Affect Transform* option, see below, or your owner has some negative scale values).

Setting equal the min and max values of an axis locks the owner's scaling along that axis. Although this is possible, using the *Transformation Properties* axis locking feature is probably easier.

Options



Minimum X ☐ 0.000
Y ☐ 0.000
Z ☐ 0.000
Maximum X ☐ 0.000
Y ☐ 0.000
Z ☐ 0.000
☐ Affect Transform
Owner Space: World Space
Influence: 1.000

Limit Scale panel.

Minimum/Maximum X, Y, Z

These buttons enable the lower boundary for the scale of the owner along respectively the X, Y and Z axes of the chosen *Space*. The *Min* and *Max* number fields to their right control the value of their lower and upper boundaries, respectively.

Note

If a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

Affect Transform

As pointed out before by default, even though visually constrained, and except for the negative values, the owner can still have scales out of bounds (as shown by the *Transform* panel). When you enable this checkbox, this is no longer possible, the owner transform properties are also limited by the constraint. However, note that, the constraint does not directly modify the scale values: you have to scale its owner one way or another for this to take effect.

Owner

This constraint allows you to choose in which space to evaluate its owner’s transform properties. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

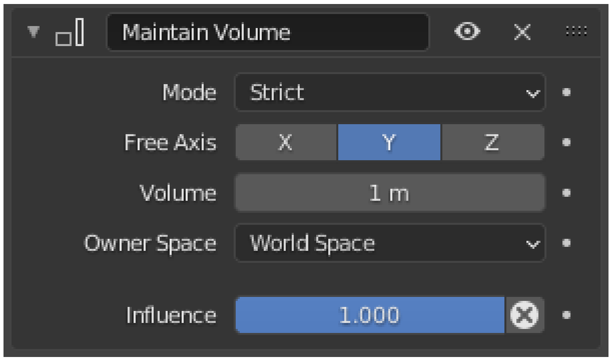
Maintain Volume Constraint

The *Maintain Volume* constraint limits the volume of a mesh or a bone to a given ratio of its original volume.

See also

[Harkyman on the development of the Maintain Volume constraint](#).

Options



Mode: Strict

Free Axis: X Y Z (Y)

Volume: 1 m

Owner Space: World Space

Influence: 1.000

Maintain Volume Constraint.

Mode

Specifies how the constraint handles scaling of the non-free axes.

Strict

This mode overrides non-free axis scaling to strictly maintain the specified volume. Only the ratio between the scale of the non-free axes is passed through.

Uniform

This mode maintains the volume as specified only when the pre-constraint scaling is uniform. Deviations from uniform scaling on non-free axes are passed through.

Single Axis

This mode maintains the volume only when the object is scaled just on its free axis. Any additional non-free axis scaling is passed through.

Free Axis

The free-scaling axis of the object.

Volume

The bone's rest volume.

Owner

This constraint allows you to choose in which space to evaluate its owner's transform properties. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

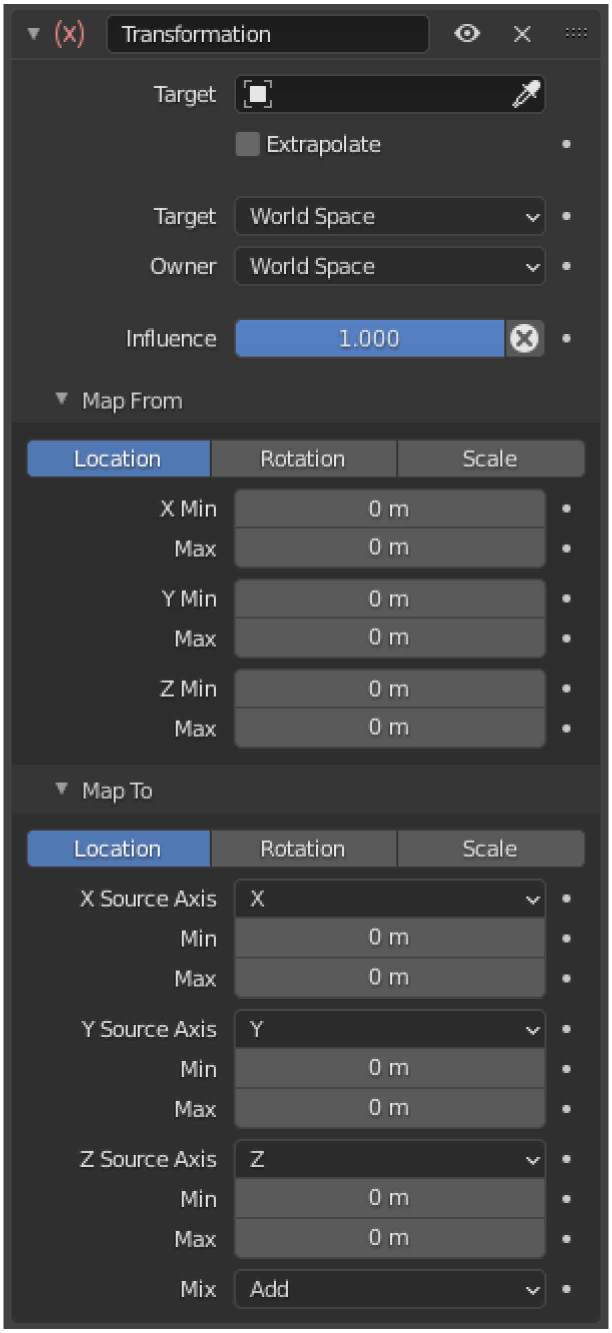
Example

Transformation Constraint

This constraint is more complex and versatile than the other “transform” constraints. It lets you set the location, rotation or scale of an object/bone based on the location, rotation or scale of another, mixing and matching axes as you see fit. An example could be to set a gear’s X rotation based on the Y coordinate of a rail next to it.

The constraint works with input and output value ranges, one for each axis. It first clamps the input value to the *Map From* range, then offsets and scales it to the corresponding *Map To* range. This lets you, say, map a Y coordinate in the range (-3m, 3m) to an X rotation in the range (0, 180°).

Options


Transformation

Target

☐ Extrapolate

Target: World Space

Owner: World Space

Influence: 1.000

Map From

Location | Rotation | Scale

X Min: 0 m
Max: 0 m
Y Min: 0 m
Max: 0 m
Z Min: 0 m
Max: 0 m

Map To

Location | Rotation | Scale

X Source Axis: X
Min: 0 m
Max: 0 m
Y Source Axis: Y
Min: 0 m
Max: 0 m
Z Source Axis: Z
Min: 0 m
Max: 0 m

Mix: Add

Transformation panel.

Target

The reference object to read a transformation property from. If you don't select one, the constraint's icon will turn red and it will have no effect.

See common constraint properties for more information.

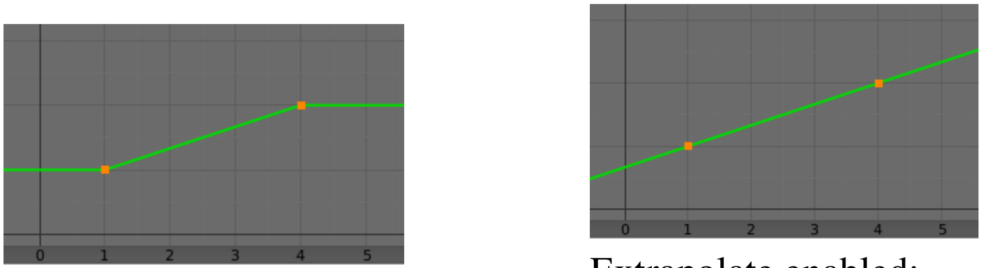
Bone

If *Target* is an Armature, you can optionally choose a bone here to use the transformation of that bone instead of the armature object as a whole.

Extrapolate

By default, the input and output values are clamped to the *Min*/*Max* values. When you enable *Extrapolate*, they’re allowed to go beyond these limits. This is illustrated with the graphs below, where the X axis represents the input (*Map From* set to *Min* = 1 and *Max* = 4) and the Y axis represents the output (*Map To* set to *Min* = 1 and *Max* = 2).

The Extrapolate option.



Extrapolate disabled: the output values are limited to the Map To range.

Extrapolate enabled: the output values can extend beyond the limits.

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of effect the constraint has on the object. See common constraint properties for more information.

Map From

The transformation to read from the *Target* (or *Bone*).

Location, Rotation, Scale
: The type of transformation to read.

Mode *Rotation*

The type of rotation to use, including different Euler orders, Quaternion, and other Rotation Channel Modes. Defaults to using the Euler order of the constraint owner.

In the *Quaternion* mode, the channels are converted to weighted angles in the same way as the swing angles of the Swing and X/Y/Z Twist modes.

X/Y/Z Min, Max

The input value range for each axis.

Map To

The transformation to apply to the owner.

Location, Rotation, Scale

The type of transformation to apply.

Order *Rotation*

Which Euler order to use. Defaults to the order of the constraint owner.

X/Y/Z Source Axis

For each of the three output axes, lets you choose the input axis that it should take its value from. You can select the same input axis multiple times.

Min, Max

The output value range for each axis.

Mix

Specifies how the result of the constraint is combined with the existing transformation. The set of available choices varies based on the type of transformation.

Replace
The result of the constraint replaces the existing transformation.

Multiply *Scale*
The new values are multiplied with the existing axis values.

Add *Location Rotation*
The new values are added to the existing axis values.

Before Original *Rotation*
The new rotation is added before the existing rotation, as if it were applied to a parent of the constraint owner.

After Original *Rotation*
The new rotation is added after the existing rotation, as if it were applied to a child of the constraint owner.

Note

- For historical reasons, the *Mix* mode defaults to *Add* for location and rotation, and *Replace* for scale.
- When using the rotation of the target as input, whatever the real values are, the constraint will always “take them back” into the (-180 to 180) range. E.g. if the target has a rotation of 420 degrees around its X axis, the values used as *X* input by the constraint will be:

  $((420 + 180) \text{ modulo } 360) - 180 = 60 - 180 = -120$

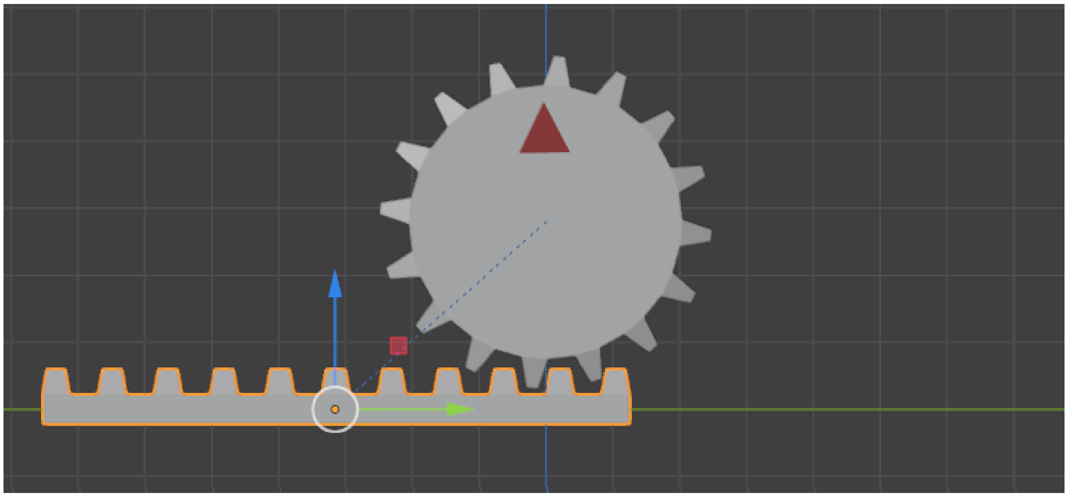
  As such, this constraint is not really suited for transforming an object based on a gear’s rotation. Rotating a gear based on an object’s transformation works fine, however.
- Similarly, when using the scale transform properties of the target as input, whatever the real values are, the constraint will always take their absolute values (i.e. invert negative ones).

- When a *Min* value is higher than its corresponding *Max* one, both are considered equal to the *Max* one. This means you cannot create “reversed” mappings.

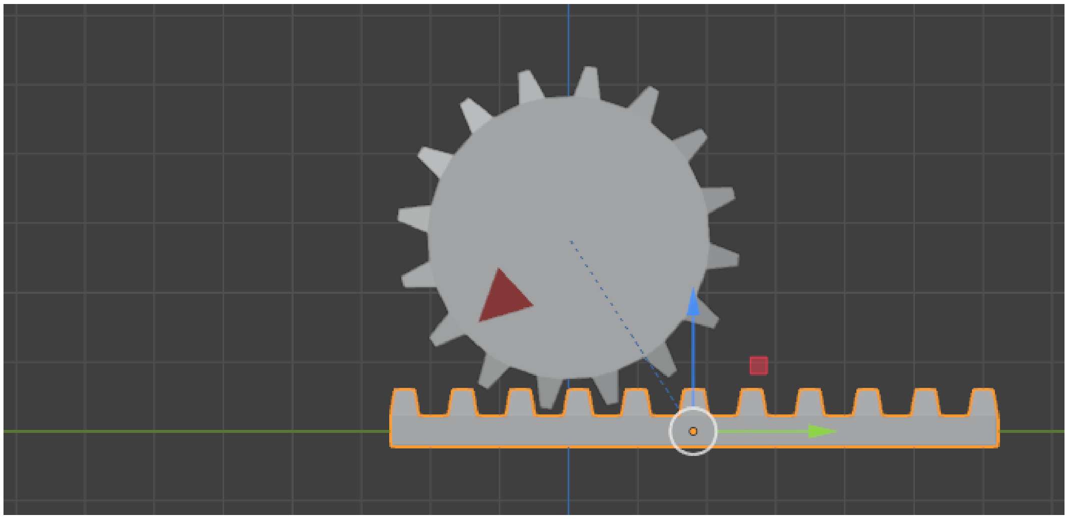
Example

In the following example, we add a constraint to a gear that sets its X rotation based on the Y position of a rail:

- Target: Rail object
- Map From: Location
  - Y Min: -3m
  - Y Max: 3m
- Map To: Rotation
  - X Source Axis: Y
  - X Min: 0°
  - X Max: 180°



Before moving the rail.



After moving the rail.

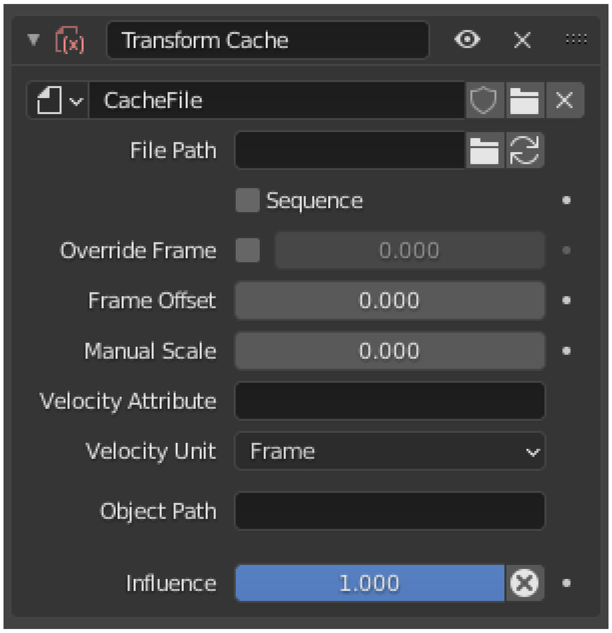
By default, the gear will stop rotating if the rail goes outside the (-3m, 3m) range. You can enable *Extrapolate* to change this.

Transform Cache Constraint

The *Transform Cache Constraint* is used to stream animations from Alembic or USD made at the transformation matrix level (for example rigid bodies, or camera movements).

When importing an Alembic or USD file, Transform Cache constraints are automatically added to objects with animated transforms. For time-varying meshes (so deforming animations), the Mesh Sequence Cache modifier is used.

Options



File Path
☐ Sequence
Override Frame ☐ 0.000
Frame Offset 0.000
Manual Scale 0.000
Velocity Attribute
Velocity Unit Frame
Object Path
Influence 1.000

Transform Cache Constraint.

Cache File

Data-block menu to select the Alembic or USD file.

File Path

Path to the Alembic or USD file.

Sequence

Whether or not the cache is separated in a series of files.

Override Frame

Whether to use a custom frame for looking up data in the cache file, instead of using the current scene frame.

The *Frame* value is the time to use for looking up the data in the cache file, or to determine which to use in a file sequence.

Frame Offset

Subtracted from the current frame to use for looking up the data in the cache file, or to determine which file to use in a file sequence.

Manual Scale

Value by which to enlarge or shrink the object with respect to the world's origin.

Velocity Attribute

The name of the Alembic attribute used for generating motion blur data; by default, this is `.velocities` which is standard for most Alembic files.

Note

The *Velocity Attribute* option is currently for Alembic files only.

Velocity Unit

Defines how the velocity vectors are interpreted with regard to time.

Frame

The velocity unit was encoded in frames and does not need to be scaled by scene FPS.

Second

The velocity unit was encoded in seconds and needs to be scaled by the scene FPS (1 / FPS).

Note

The *Velocity Unit* option is currently for Alembic files only.

Object Path

The path to the Alembic or USD object inside the archive or stage.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Tracking Constraints

- Clamp To Constraint
  - Options
  - Example
- Damped Track Constraint
  - Options
  - Example
- Inverse Kinematics Constraint
  - Options
  - Example
- Locked Track Constraint
  - Options
  - Example
- Spline IK Constraint
  - Options
  - Example
- Stretch To Constraint
  - Options
  - Example
- Track To Constraint
  - Options
  - Example

Clamp To Constraint

The *Clamp To* constraint clamps an object to a curve. The *Clamp To* constraint is very similar to the Follow Path constraint, but instead of using the evaluation time of the target curve, *Clamp To* will get the actual location properties of its owner (those shown in the *Transform* panel), and judge where to put it by “mapping” this location along the target curve.

One benefit is that when you are working with *Clamp To*, it is easier to see what your owner will be doing; since you are working in the 3D Viewport, it will just be a lot more precise than sliding keys around on an F-Curve and playing the animation over and over.

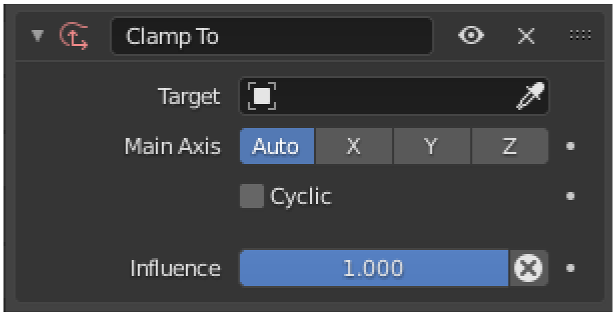
A downside is that unlike in the Follow Path constraint, *Clamp To* does not have any option to track your owner’s rotation (pitch, roll, yaw) to the banking of the targeted curve, but you do not always need rotation on, so in cases like this it’s usually a lot handier to fire up a *Clamp To*, and get the bits of rotation you do need some other way.

The mapping from the object’s original position to its position on the curve is not perfect, but uses the following simplified algorithm:

- A “main axis” is chosen, either by the user, or as the longest axis of the curve’s bounding box (the default).
- The position of the object is compared to the bounding box of the curve in the direction of the main axis. So for example if X is the main axis, and the object is aligned with the curve bounding box’s left side, the result is 0; if it is aligned with the right side, the result is 1.
- If the cyclic option is unchecked, this value is clamped in the range 0-1.
- This number is used as the curve time, to find the final position along the curve that the object is clamped to.

This algorithm does not produce exactly the desired result because curve time does not map exactly to the main axis position. For example an object directly in the center of a curve will be clamped to a curve time of 0.5 regardless of the shape of the curve, because it is halfway along the curve's bounding box. However, the 0.5 curve time position can actually be anywhere within the bounding box!

Options



Influence 1.000

Clamp To panel.

Target

The target [Data ID](#) indicates which curve object the Clamp To constraint will track along. The target must be a curve object type. See [common constraint properties](#) for more information.

Main Axis

This button group controls which global axis (X, Y or Z) is the main direction of the path. When clamping the object to the target curve, it will not be moved significantly on this axis. It may move a small amount on that axis because of the inexact way this constraint functions.

For example if you are animating a rocket launch, it will be the Z axis because the main direction of the launch path is up. The default *Auto* option chooses the axis which the curve is longest in (or X if they are equal). This is usually the best option.

Cyclic

By default, once the object has reached one end of its target curve, it will be constrained there. When the *Cyclic* option is enabled, as soon as it reaches one end of the curve, it is instantaneously moved to its other end. This is of course primarily designed for closed curves (e.g. circles), as this allows your owner to go around it over and over.

Influence

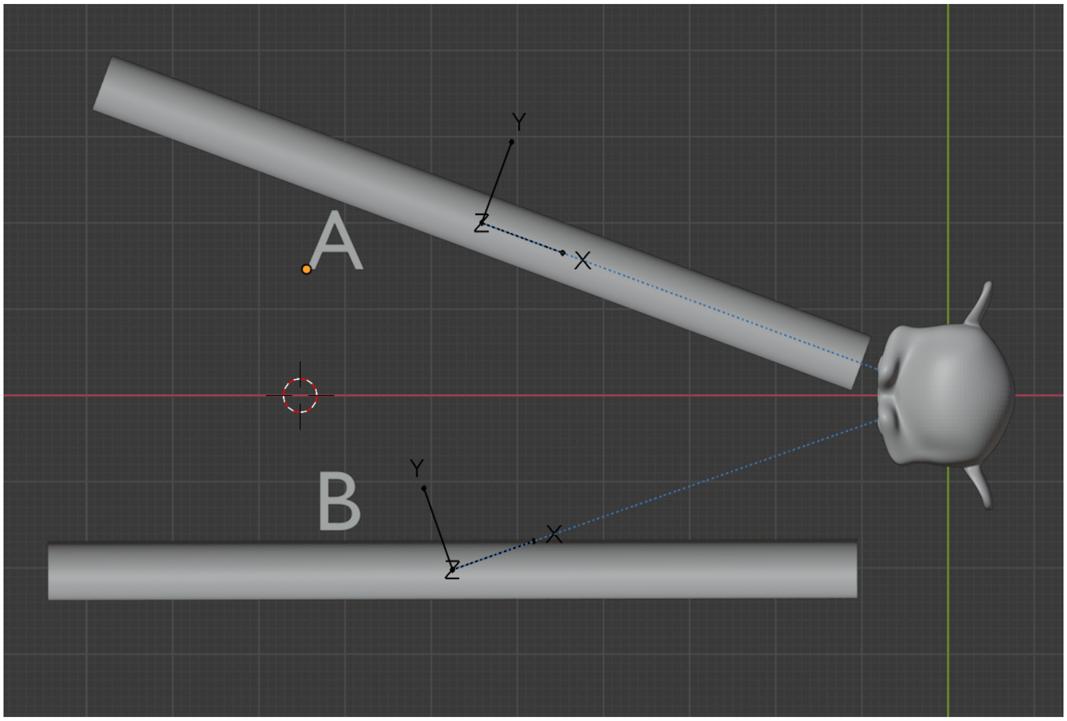
Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Damped Track Constraint

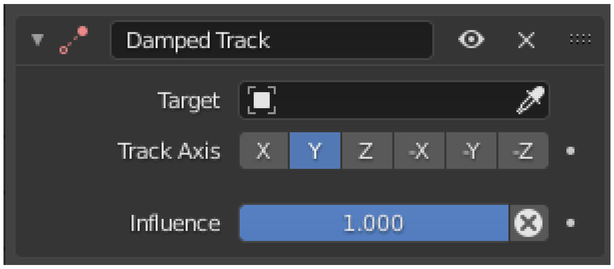
The *Damped Track* constraint constrains one local axis of the owner to always point towards *Target*. This constraint uses a pure Swing rotation, i.e. the shortest possible single axis rotation. In other 3D software you can find it with the name “Look at” constraint.

Although usually associated with bones, Damped Track can align objects to point to (and follow) other objects or bones. It is important to note that the constraint aligns the origin’s axes to point to the target’s origin point. This is illustrated in the following figure. In each case the objects are set as Damped Track to +X.



A: Object vertices aligned along axis origin. B: Object vertices aligned away from origin.

Options



Influence 1.000

Damped Track panel.

Target

Data ID used to select the constraint's target, and is not functional (red state) when it has none. See common constraint properties for more information.

Track Axis

Once the owner object has had a Damped Track constraint applied to it, you must then choose which axis of the object you want to point at the Target object. The negative axis direction cause the object to point away from the Target object along the selected axis direction.

-X, -Y, -Z, X, Y, Z

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Inverse Kinematics Constraint

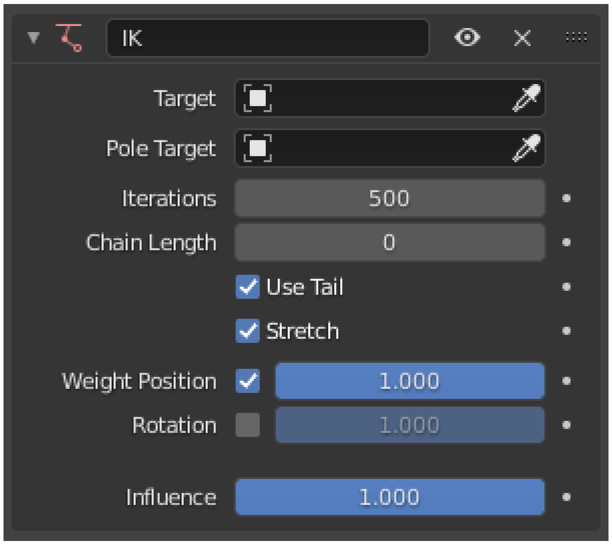
The *Inverse Kinematics* constraint implements the *inverse kinematics* armature posing technique. Hence, it is only available for bones. To quickly create an IK constraint with a target, select a bone in Pose Mode, and press `Shift-I`.

This constraint is fully documented in the Inverse Kinematics page, part of the rigging chapter.

Note

The IK constraints are special in that they modify multiple bones. For this reason, they ignore their position in the stack and always run after all other constraints on the affected bones. To apply constraints after IK, it is necessary to first copy the final transformation to a new bone chain, e.g. using Copy Transforms.

Options



IK

Target

Pole Target

Iterations 500

Chain Length 0

☑ Use Tail

☑ Stretch

Weight Position ☑ 1.000

Rotation ☐ 1.000

Influence 1.000

Inverse Kinematics panel.

Target
: Data ID used to select the an armature. See common constraint properties for more information.

Pole Target
: Object for pole rotation.

Iterations
: Maximum number of solving iterations.

Chain Length
: How many bones are included in the IK effect. Set to 0 to include all bones.

Use Tail
: Include bone’s tail as last element in chain.

Stretch
: Enable IK stretching.

Weight Position
: For Tree-IK: Weight of position control for this target.

Rotation
: Chain follows rotation of target.

Target
: Disable for targetless IK.

Influence
: Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

iTaSC Solver

If the iTaSC IK Solver is used, the IK Solver Constraint changes to add these addition parameters.

IK Type

- Copy Pose
  - Equivalent to the traditional end effector position and orientation constraint: the end effector is constrained to take the position, and optionally the orientation, of a given target, which is set in the target field.
  - Position/Rotation Locking
    - Allows to obtain various effect by not constraining the coordinates along certain axis.
    - Axis Reference
      - Specifies how to compute the axis coordinates.
      - Bone
        - The coordinates are the position and orientation of the target relative to the bone.
      - Target
        - The opposite of *Bone*, the coordinates are the position and orientation of the tip of the bone relative to the target.
- Distance
  - Specify that the end effector will stay inside, at, or outside a sphere centered on the target object.
  - Limit Mode
    - Inside
      - The end effector will stay inside of the distance from the target object.
    - Outside

The end effector will stay outside of the distance from the target object.

On Surface

The end effector will stay exactly at the distance from the target object.

Distance

The radius from the target object.

Note

The *Influence* parameter is not implemented if *Pole Target* is used.

Example

Locked Track Constraint

The *Locked Track* constraint is a bit tricky to explain, both graphically and textual. Basically, it is a Track To constraint, but with a locked axis, i.e. an axis that cannot rotate (change its orientation). Hence, the owner can only track its target by rotating around this axis, and unless the target is in the plane perpendicular to the locked axis, and crossing the owner, this owner cannot really point at its target.

Let us take the best real-world equivalent: a compass. It can rotate to point in the general direction of its target (the magnetic North, or a neighbor magnet), but it cannot point *directly at it*, because it spins like a wheel on an axle. If a compass is sitting on a table and there is a magnet directly above it, the compass cannot point to it. If we move the magnet more to one side of the compass, it still cannot point *at* the target, but it can point in the general direction of the target, and still obey its restrictions of the axle.

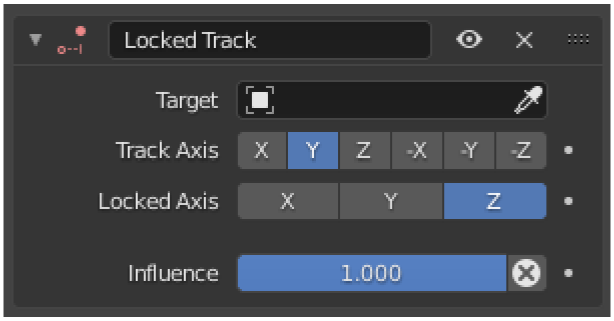
When using a *Locked Track* constraint, you can think of the target as a magnet, and the owner as a compass. The *Lock* axis will function as the axle around which the owner spins, and the *To* axis will function as the compass' needle. Which axis does what is up to you!

If you have trouble understanding the buttons of this constraint, read the tooltips, they are pretty good. If you do not know where your object's axes are, enable *Axis* in *Properties* ‣ *Armature* ‣ *Viewport Display*. Or, if you are working with bones, turn on the *Axes* in the armature's *Viewport Display* panel.

This constraint was designed to work cooperatively with the *Track To* constraint. If you set the axes buttons right for these two constraints, *Track To* can be used to point the axle at a primary target, and *Locked Track* can spin the owner around that axle to a secondary target.

This constraint also works very well for 2D billboarding.

Options



Locked track panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Track Axis

The tracking local axis, i.e. the owner's axis to point at the target. The negative options force the relevant axis to point away from the target.

Locked Axis

The locked local axis, i.e. the owner's axis which cannot be re-oriented to track the target.

Important

If you choose the same axis for *Tracking Axis* and *Locked Axis*, the constraint will no longer be functional (red state).

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Spline IK Constraint

The *Spline IK* constraint aligns a chain of bones along a curve. By leveraging the ease and flexibility of achieving aesthetically pleasing shapes offered by curves and the predictability and well-integrated control offered by bones, *Spline IK* is an invaluable tool in the riggers' toolbox. It is particularly well suited for rigging flexible body parts such as tails, tentacles, and spines, as well as inorganic items such as ropes.

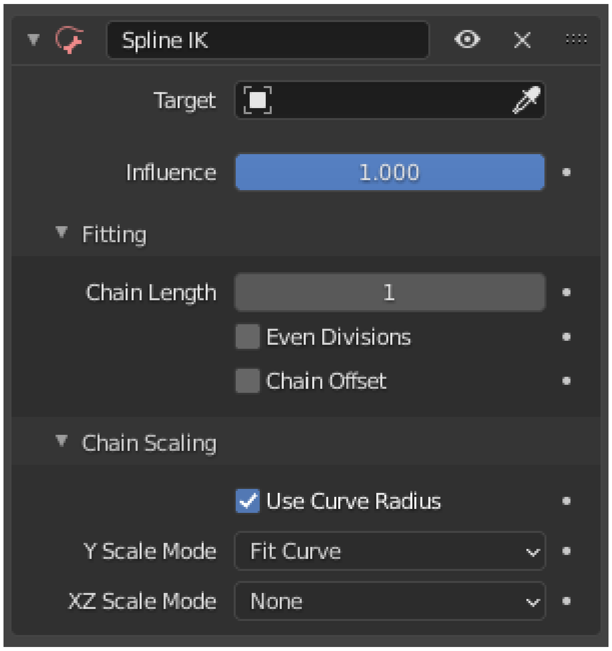
To set up *Spline IK*, it is necessary to have a chain of connected bones and a curve to constrain these bones to:

- With the last bone in the chain selected, add a *Spline IK* constraint from the *Bone Constraints* tab in the *Properties*.
- Set the 'Chain Length' setting to the number of bones in the chain (starting from and including the selected bone) that should be influenced by the curve.
- Finally, set *Target* to the curve that should control the curve.

Note

The IK constraints are special in that they modify multiple bones. For this reason, they ignore their position in the stack and always run after all other constraints on the affected bones. To apply constraints after IK, it is necessary to first copy the final transformation to a new bone chain, e.g. using Copy Transforms.

Options


Spline IK
Target
Influence 1.000
Fitting
Chain Length 1
☐ Even Divisions
☐ Chain Offset
Chain Scaling
☑ Use Curve Radius
Y Scale Mode: Fit Curve
XZ Scale Mode: None

Spline IK panel.

Target

Data ID used to select the target curve. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Fitting

Chain Length

How many bones are included in the chain.

Even Division

Ignore the relative length of the bones when fitting to the curve.

Chain Offset

Retain the offset of the root joint from the start point of the curve.

Chain Scaling

Use Curve Radius

Average radius of the endpoints is used to tweak the X and Z scaling of the bones, on top of the X and Z scale mode.

Y Scale Mode

Specifies how the length of a bone is scaled when it is fitted to the curve, in addition to the effects of target curve object scale and curvature.

None

The bone is reset to its rest pose length.

Fit Curve

The bones are stretched to cover the entire length of the curve.

Bone Original

The original Y axis scale of the bone is used.

XZ Scale Mode

Scaling that a bone undergoes in the other two directions.

None

Do not scale the X and Z axes.

Bone Original

Use the original scaling of the bones.

Inverse Scale

Scale of the X and Z axes is the inverse of the Y scale.

Volume Preservation

Similar to the Stretch To constraint.

Use Original Scale

Specifies that *Inverse Scale* or *Volume Preservation* should be applied on top of the original scaling of the bones, like in the Stretch To constraint.

See also

This subject is seen in-depth in the Armature Posing section.

Example

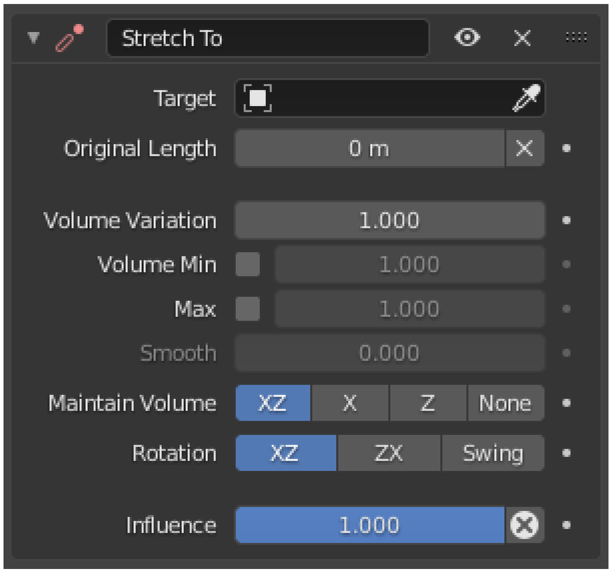
Stretch To Constraint

The *Stretch To* constraint causes its owner to rotate and scale its Y axis towards its target. So it has the same tracking behavior as the Track To constraint. However, it assumes that the Y axis will be the tracking and stretching axis, and does not give you the option of using a different one.

It also optionally has some raw volumetric features, so the owner can squash down as the target moves closer, or thin out as the target moves farther away. Note however, that it is not the real volume of the owner which is thus preserved, but rather the virtual one defined by its scale values. Hence, this feature works even with non-volumetric objects, like empties, 2D meshes or surfaces, and curves.

With bones, the “volumetric” variation scales them along their own local axes (remember that the local Y axis of a bone is aligned with it, from root to tip).

Options



Stretch To

Target	
Original Length	0 m
Volume Variation	1.000
Volume Min	☐ 1.000
Max	☐ 1.000
Smooth	0.000
Maintain Volume	[✓ XZ] X Z None
Rotation	[✓ XZ] ZX Swing
Influence	1.000

Stretch To panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Original Length

This number field sets the rest distance between the owner and its target, i.e. the distance at which there is no deformation (stretching) of the owner.

(Reset Original Length)

When clicked, this small button will recalculate the *Rest Length* value, so that it corresponds to the actual distance between the owner and its target (i.e. the distance before this constraint is applied).

Volume Variation

This number field controls the amount of "volume" variation exponentially to the stretching amount. Note that the 0.0 value is not allowed, if you want to disable the volume feature, use the *None* button (see below).

Volume Min, Max

Limits for the volume preservation to a minimum and maximum scaling each by a *Bulge* factor.

Smooth

Smoothness factor to make limits less visible.

Maintain Volume

These buttons control which of the X and/or Z axes should be affected (scaled up/down) to preserve the virtual volume while stretching along the Y axis. If you enable the *none* button, the volumetric features are disabled.

Rotation

Specifies how the owner should be rotated to track the target with its Y axis.

XZ, ZX

These buttons are equivalent to the *Up* ones of the Track To constraint: the owner is rotated around two local axes in the specified order.

Swing

The constraint uses a single Swing rotation, equivalent to the Damped Track constraint.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Track To Constraint

The *Track To* constraint applies rotations to its owner, so that it always points a given "To" axis towards its target, with another "Up" axis permanently maintained as much aligned with the global Z axis (by default) as possible. This tracking is similar to the "billboard tracking" in 3D (see note below).

This is the preferred tracking constraint, as it has a more easily controlled constraining mechanism.

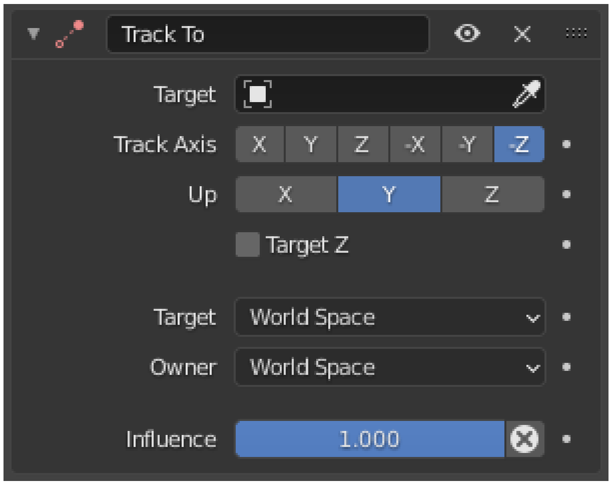
This constraint shares a close relationship to the Inverse Kinematics constraint in some ways.

Tip

Billboard Tracking

The term "billboard" has a specific meaning in real-time CG programming (i.e. video games!), where it is used for plane objects always facing the camera (they are indeed "trackers", the camera being their "target"). Their main usage is as support for tree or mist textures: if they were not permanently facing the camera, you would often see your trees squeezing to nothing, or your mist turning into a mille-feuille paste, which would be funny but not so credible.

Options


Track To

Target

Track Axis: X Y Z -X -Y [✓ -Z]

Up: X [✓ Y] Z

☐ Target Z

Target: World Space

Owner: World Space

Influence: 1.000

Track To panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Track Axis

The tracking local axis, i.e. the owner's axis to point at the target. The negative options force the relevant axis to point away from the target.

Up

The "upward-most" local axis, i.e. the owner's axis to be aligned (as much as possible) with the global Z axis (or target Z axis, when the *Target* button is enabled).

Target Z

By default, the owner's *Up* axis is (as much as possible) aligned with the global Z axis, during the tracking rotations. When this button is enabled, the *Up* axis will be (as much as possible) aligned with the target's local Z axis?

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Warning

If you choose the same axis for *Tracking Axis* and *Up*, the constraint will not be functional anymore (red state).

Example

Relationship Constraints

- Action Constraint
    - Options
    - Notes
    - Example
- Armature Constraint
    - Options
- Child Of Constraint
    - Options
    - Tips
    - Technical Note
    - Examples
- Floor Constraint
    - Options
    - Example
- Follow Path Constraint
    - Options
    - Example
- Pivot Constraint
    - Options
    - Example
- Shrinkwrap Constraint
    - Options
    - Example

Action Constraint

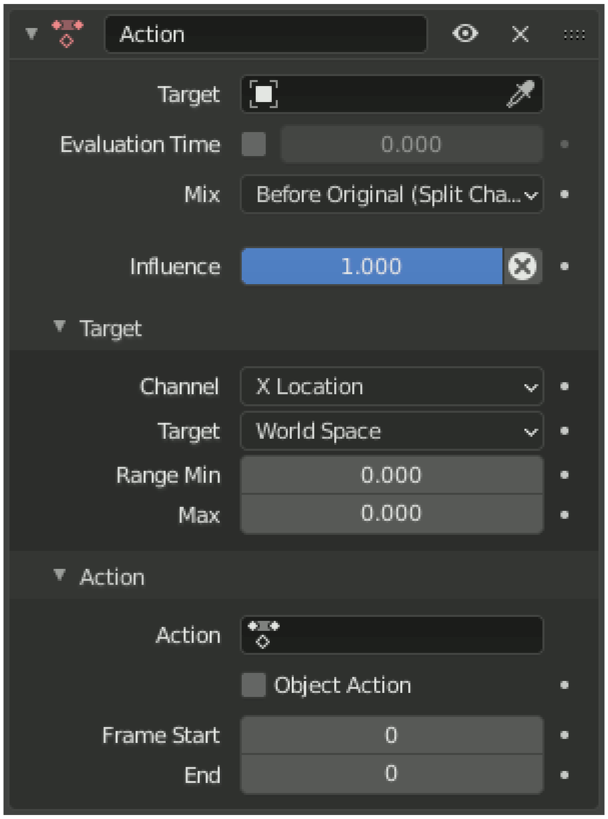
The *Action* constraint is powerful. It allows you control an Action using the transformations of another object.

The underlying idea of the *Action* constraint is very similar to the one behind the Drivers, except that the former uses a whole action (i.e. multiple F-Curves of the same type), while the latter controls a single F-Curve of their “owner”…

Note that even if the constraint accepts the *Mesh* action type, only the *Object*, *Pose* and *Constraint* types are really working, as constraints can only affect objects’ or bones’ transform properties, and not meshes’ shapes. Also note that only the object transformation (location, rotation, scale) is affected by the action, if the action contains keyframes for other properties they are ignored, as constraints do not influence those.

As an example, let us assume you have defined an *Object* action (it can be assigned to any object, or even no object at all), and have mapped it on your owner through an *Action* constraint, so that moving the target in the (0.0 to 2.0) range along its X axis maps the action content on the owner in the (0 to 100) frame range. This will mean that when the target’s *X* property is 0.0 the owner will be as if in frame 0 of the linked action; with the target’s *X* property at 1.0 the owner will be as if in frame 50 of the linked action, etc.

Options



Action

Target

Evaluation Time ☐ 0.000

Mix Before Original (Split Cha...

Influence 1.000

Target

Channel X Location

Target World Space

Range Min 0.000

Max 0.000

Action

Action

☐ Object Action

Frame Start 0

End 0

Action panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Evaluation Time

This property allows objects to be driven without a constraint target by interpolating between the *Action Start* and *End* frames. The relative position between the start and end frame can be controlled using the value slider.

This is very helpful for more complex rigging and mechanical rigs, as it means the Action constraint can be controlled directly with a Driver or Custom Property.

Mix

Specifies how the keyframed transformation from the action is combined with the existing transformation. These modes are the same as in the Copy Transforms constraint.

Before/After Original (Full)

The keyframed transformation is added before/after the existing transformation, as if it was applied to an imaginary parent/child of the constraint owner. Scale is handled like in the most basic Full Inherit Scale mode of bones, so combining non-uniform scale and rotation will create shear.

Before/After Original (Aligned)

The keyframed transformation is added before/after the existing transformation, as if it was applied to an imaginary parent/child of the constraint owner. Scale is handled like in the Aligned Inherit Scale mode of bones to avoid creating shear.

This is equivalent to using the *Split Channels* option, but replacing the location component with the result of *Full*. If only uniform scale is used, the result is identical to *Full*.

Before/After Original (Split Channels)

Combines location, rotation and scale components of the transformation separately, similar to a sequence of three Copy Location, Copy Rotation and Copy Scale (with Offset) constraints bundled together in one operation; the result may be slightly different in case of sheared inputs.

Unlike *Aligned*, in this mode location channels are simply added together, so rotation and scale components of the input transformations cannot affect the resulting location.

Warning

For technical reasons modes other than *After Original (Full)* and *After Original (Aligned)* may not work as expected for constraints on *objects* (not bones) without a parent.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Target

Channel

This selector controls which transform property (location, rotation or scale along/around one of its axes) from the target to use as “action driver”.

Target

This constraint allows you to choose in which space to evaluate its target’s transform properties.

Range Min, Max

The lower and upper bounds of the driving transform property value.

Warning

Unfortunately, here again we find the constraint’s limitations:

- When using a rotation property as “driver”, these values are “mapped back” to the (-180.0 to 180.0) range.
- When using a scale property as “driver”, these values are limited to null or positive values.

Action

Action

Select the name of the action you want to use.

Warning

Even though it might not be in red state (UI refresh problems…), this constraint is obviously not functional when this field does not contain a valid action.

Object Action

Bones **only**, when enabled, this option will make the constrained bone use the “object” part of the linked action, instead of the “same-named pose” part. This allows you to apply the action of an object to a bone.

Frame Start, End

The starting and ending frames of the action to be mapped.

Note

- These values must be strictly positive.
- By default, both values are set to 0, which disables the mapping (i.e. the owner just gets the properties defined at frame 0 of the linked action…).

Notes

- When the object or bone already has Action constraints, the next constraint using a newly keyframed action should be added before all others in order to get the same final combined transformation. This fact is not affected by the Mix mode.
- Unlike usual, you can have a *Start* value higher than the *End* one, or a *Min* one higher than a *Max* one: this will reverse the mapping of the action (i.e. it will be “played” reversed…), unless you have both sets reversed, obviously!
- When using a *Constraint* action, it is the constraint *channel’s names* that are used to determine to which constraints of the owner apply the action. E.g. if you have a constraint channel named “trackto_empt1”, its keyed *Influence* and/or *Head/Tail* values (the only ones you can key) will be mapped to the ones of the owner’s constraint named “trackto_empt1”.

- Similarly, when using a *Pose* action (which is obviously only meaningful and working when constraining a bone!), it is the bone's name that is used to determine which bone *channel's names* from the action to use (e.g. if the constrained bone is named "arm", it will use and only use the action's bone channel named "arm"…). Unfortunately, using a *Pose* action on a whole armature object (to affect all the keyed bones in the action at once) will not work…
- Actions can also be marked as Asset, but with certain limitations. For more info, see Pose Library.

Example

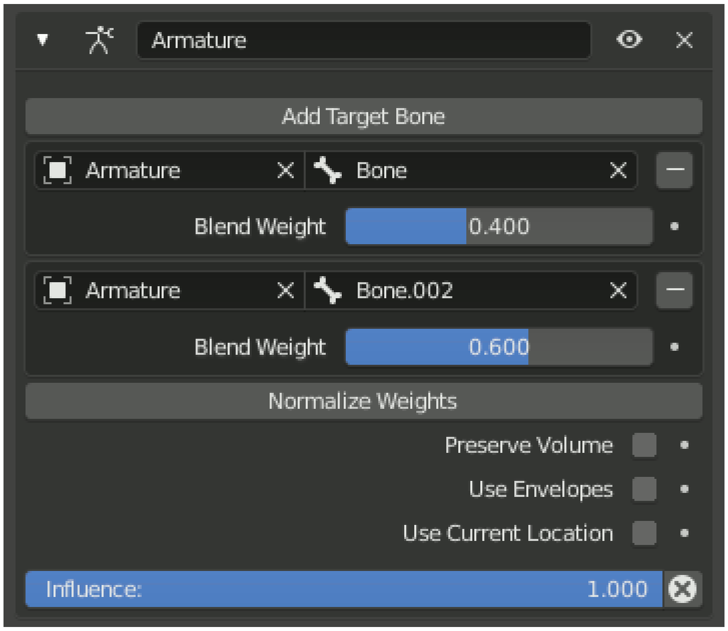
Armature Constraint

Armature is the constraint version of the Armature Modifier, exactly reproducing the weight-blended bone transformations and applying it to its owner orientation. It can be used like a variant of the Child Of constraint that can handle multiple parents at once, but requires all of them to be bones.

Note

Unlike the Armature modifier, the constraint does not take the Deform checkbox of bones into account, and can use any bone as target.

Options



Armature
Armature | Bone
Blend Weight 0.400
Armature | Bone.002
Blend Weight 0.600
Preserve Volume ☐
Use Envelopes ☐
Use Current Location ☐
Influence: 1.000

Armature constraint.

Preserve Volume

Like the matching option of the modifier, it enables the use of quaternions for preserving the volume of the object during deformation.

Use Envelopes

To approximate envelope-only behavior of the modifier, add all relevant bones with weight 1.0 and enable this option.

Note

Unlike the modifier, the constraint always requires explicitly listing all of its target bones with associated weights. This option merely enables envelopes for all bones, as if they had Envelope Multiply enabled.

Use Current Location

Only for constraints on bones: Instead of using the rest location, use the current location of the owner bone to compute envelope weights or binding to B-Bone segments.

With envelope weights, this can be used to change the active “parent” bone of the owner bone dependent on its location. For non-bones this mode is always active, because they don’t have a rest location.

Add Target Bone

This button adds a new empty entry at the end of the target list.

Normalize Weights

This button normalizes all weight values in the target list so that they add up to 1.0.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Bones

This specifies the list of bones used by the constraint to deform its owner. Each target bone has the following input fields and controls:

Target
: Unlike the modifier, the constraint can use bones coming from different armatures at the same time. See common constraint properties for more information.

Sub-target
: Name of the target bone.

(Remove Target)
: Removes the entry from the target list.

Weight
: Weight associated with the bone, equivalent to vertex groups in the modifier.

Child Of Constraint

Child Of is the constraint version of the standard parent/children relationship between objects (the one established through the `Ctrl-P` shortcut, in the 3D Viewport).

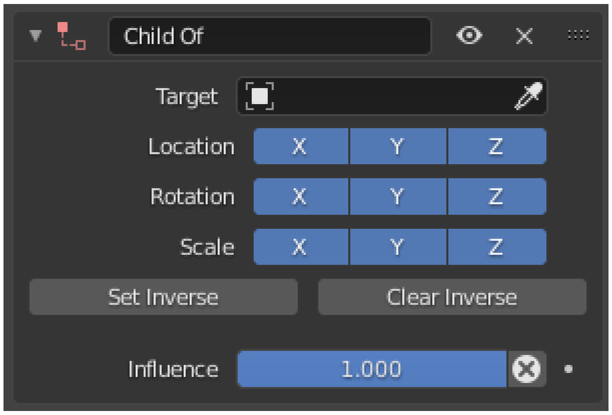
Parenting with a constraint has several advantages and enhancements, compared to the traditional method:

- You can have several different parents for the same object (weighting their respective influence with the *Influence* slider).
- As with any constraint, you can key (i.e. animate) its Influence setting. This allows the object which has a Child Of constraint upon it to change over time which target object will be considered the parent, and therefore have influence over it.

Important

Do not confuse this "basic" object parenting with the one that defines the chains of bones inside of an armature. This constraint is used to parent an object to a bone (the so-called object skinning), or even bones to bones. But do not try to use it to define chains of bones.

Options



Influence 1.000

Child Of panel.

Target

The target object that this object will act as a child of. Data ID used to select the constraint's target, and is not functional (red state) when it has none. See common constraint properties for more information.

Location

Each of these buttons will make the parent affect or not affect the location along the corresponding axis.

Rotation

Each of these buttons will make the parent affect or not affect the rotation around the corresponding axis.

Scale

Each of these buttons will make the parent affect or not affect the scale along the corresponding axis.

Set Inverse

By default, when you parent your owner to your target, the target becomes the origin of the owner's space. This means that the location, rotation and scale of the owner are offset by the same properties of the target. In other words, the owner is transformed when you parent it to your target. This might not be desired! So, if you want to restore your owner to its before-parenting state, click on the *Set Inverse* button.

Clear Inverse

This button reverses (cancels) the effects of the above one, restoring the owner/child to its default state regarding its target/parent.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Tips

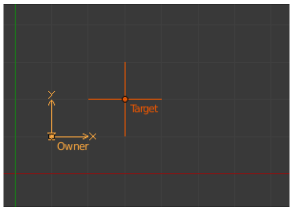
When creating a new parent relationship using this constraint, it is usually necessary to click on the *Set Inverse* button after assigning the parent. As noted above, this cancels out any unwanted transform from the parent, so that the owner returns to the location/rotation/scale it was in before the constraint was applied. Note that you should apply *Set Inverse* with all other constraints disabled (their *Influence* set to 0.0) for a particular *Child Of* constraint, and before transforming the target/parent (see example below).

About the toggle buttons that control which target's (i.e. parent's) individual transform properties affect the owner, it is usually best to leave them all enabled, or to disable all three of the given Location, Rotation and Scale transforms.

Technical Note

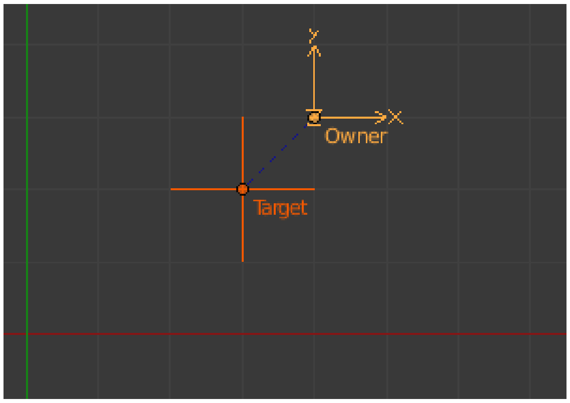
If you use this constraint with all channels on, it will use a straight matrix multiplication for the parent relationship, not decomposing the parent matrix into loc/rot/size. This ensures any transformation correctly gets applied, also for combinations of rotated and non-uniform scaled parents.

Examples


Target
Owner
Y
X

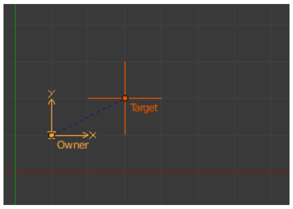
No constraint.

Note the position of Owner empty 1.0 unit along the X and Y axes.


Y
X
Owner
Target

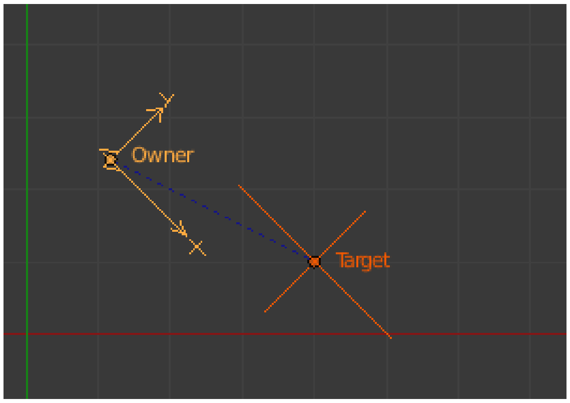
Child Of just added.

Here you can see that Owner empty is now 1.0 unit away from Target_1 empty along X and Y axes.


Target
Owner
Y
X

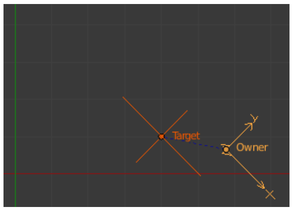
Offset set.

Set Inverse has been clicked, and Owner is back to its original position.


Y
Owner
X
Target

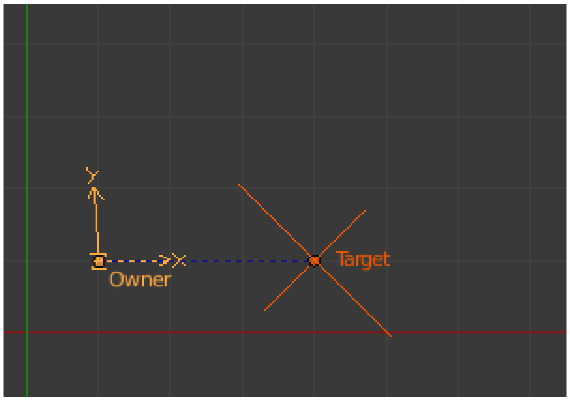
Target/parent transformed.

Target_1 has been moved along the XY plane, rotated around the Z axis, and scaled along its local X axis.



Offset cleared.

Clear Inverse has been clicked. Owner is fully again controlled by Target_1.



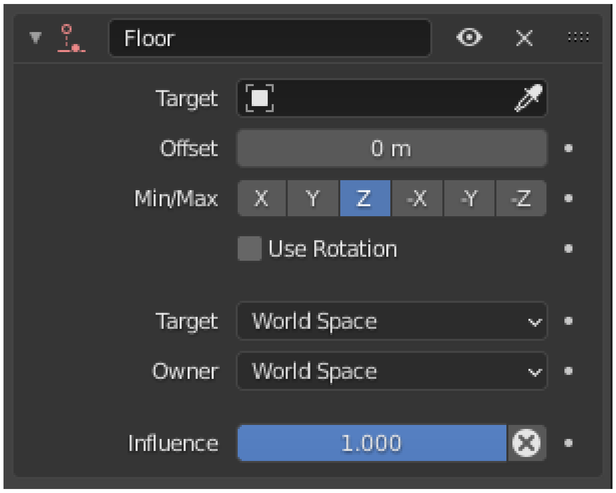
Offset set again.

Set Offset has been clicked again. As you can see, it does not gives the same result as in (Target/parent transformed). As noted above, use Set Inverse only once, before transforming your target/parent.

Floor Constraint

The *Floor* constraint allows you to use its target position (and optionally rotation) to specify a plane with a “forbidden side”, where the owner cannot go. This plane can have any orientation you like. In other words, it creates a floor (or a ceiling, or a wall)! Note that it is only capable of simulating entirely flat planes, even if you use the *Vertex Group* option. It cannot be used for uneven floors or walls.

Options



Floor

Target

Offset: 0 m

Min/Max: X Y Z -X -Y -Z (Z selected)

☐ Use Rotation

Target: World Space

Owner: World Space

Influence: 1.000

Floor panel.

Target

Data ID used to select the constraints target, and is not functional (red state) when it has none. See common constraint properties for more information.

Offset

Allows you to offset the “floor” plane from the target’s origin, by the given number of units. Use it e.g. to account for the distance from a foot bone to the surface of the foot’s mesh.

Max/Min

Controls which plane will be the “floor”. The names of the buttons correspond, indeed, to the *normal* to this plane (e.g. enabling Z means “XY plane”, etc.). By default, these normals are aligned with the *global* axes. However, if you enable *Use Rotation* (see above), they will be aligned with the *local target’s axes*. As the constraint does not only define an uncrossable plane, but also a side of it which is forbidden to the owner, you can choose which side by enabling either the positive or negative normal axis… e.g. by default Z, the owner is stuck in the positive Z coordinates.

Use Rotation

Forces the constraint to take the target’s rotation into account. This allows you to have a “floor” plane of any orientation you like, not just the global XY, XZ and YZ ones…

Target/Owner

Standard conversion between spaces. See common constraint properties for more information.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Follow Path Constraint

The *Follow Path* constraint places its owner onto a *curve* target object, and makes it move along this curve (or path). It can also affect its owner's rotation to follow the curve's bends, when the *Follow Curve* option is enabled.

It could be used for complex camera traveling, a train on its rails and most other vehicles can also use "invisible" tracks, the links of a bicycle chain, etc.

The owner is always evaluated in the global (world) space:

- Its location (as shown in the *Transform* panel) is used as an offset from its normal position on the path. E.g. if you have an owner with the (1.0, 1.0, 0.0) location, it will be one unit away from its normal position on the curve, along the X and Y axis. Hence, if you want your owner *on* its target path, clear its location `Alt-G`!
- This location offset is also proportionally affected by the scale of the target curve. Taking the same (1.0, 1.0, 0.0) offset as above, if the curve has a scale of (2.0, 1.0, 1.0), the owner will be offset two units along the X axis (and one along the Y one)…
- When the *Follow Curve* option is enabled, its rotation is also offset to the one given by the curve. E.g. if you want the Y axis of your object to be aligned with the curve's direction, it must be in rest, non-constrained state, aligned with the global Y axis. Here again, clearing your owner's rotation `Alt-R` might be useful…

The movement of the owner along the target curve/path may be controlled in two different ways:

- The most simple is to define the number of frames of the movement, in the Path Animation panel of the Curve tab, via the *Frames* number field, and its start frame via the constraint's Offset option (by default, start frame: 1 [= offset of 0], duration: 100).

- The second way, much more precise and powerful, is to define an *Evaluation Time* interpolation curve for the *Target* path (in the *Graph Editor*). See the Graph Editor chapter to learn more about F-Curves.
- If you do not want your owner to move along the path, you can give to the target curve a flat *Speed* F-Curve (its value will control the position of the owner along the path).

Follow Path is another constraint that works well with the Locked Track one. One example is a flying camera on a path. To control the camera's roll angle, you can use a *Locked Track* and a target object to specify the up direction, as the camera flies along the path.

Note

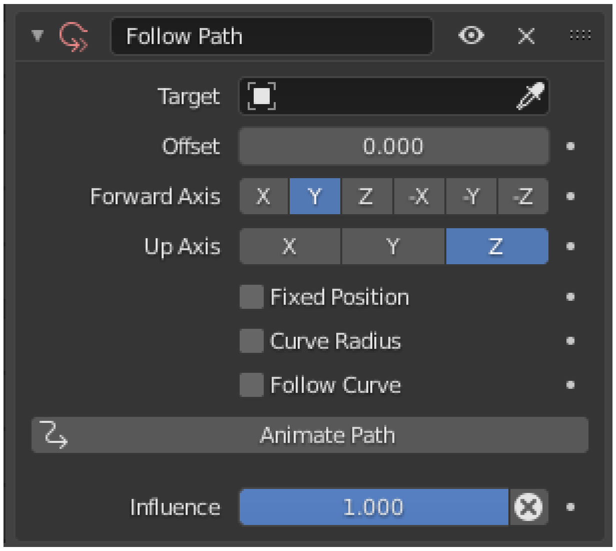
Follow Path & Clamp To

Do not confuse these two constraints. Both of them constraint the location of their owner along a curve, but *Follow Path* is an "animation-only" constraint, inasmuch as the position of the owner along the curve is determined by the time (i.e. current frame), whereas the Clamp To *constraint* determines the position of its owner along the curve using one of its location properties' values.

Note

Note that you also need to keyframe Evaluation Time for the Path. Select the path, go to the *Path Animation* panel in the curve properties, set the overall frame to the first frame of the path (e.g. frame 1), set the value of Evaluation time to the first frame of the path (e.g. 1), right-click on Evaluation time, select create keyframe, set the overall frame to the last frame of the path (e.g. frame 100), set the value of Evaluation time to the last frame of the path (e.g. 100), right-click on Evaluation time, select create keyframe.

Options



Follow Path

Target

Offset 0.000

Forward Axis X [✓ Y] Z -X -Y -Z

Up Axis X Y [✓ Z]

☐ Fixed Position

☐ Curve Radius

☐ Follow Curve

Animate Path

Influence 1.000

Follow Path panel.

Target

Data ID used to select the constraint's target, which *must* be a curve object, and is not functional (red state) when it has none. See common constraint properties for more information.

Offset

The number of frames to offset from the "animation" defined by the path (by default, from frame 1).

Forward Axis

The axis of the object that has to be aligned with the forward direction of the path (i.e. tangent to the curve at the owner's position).

Up Axis

The axis of the object that has to be aligned (as much as possible) with the world Z axis. In fact, with this option activated, the behavior of the owner shares some properties with the one caused by a Locked Track constraint, with the path as "axle", and the world Z axis as "magnet".

Fixed Position

Object will stay locked to a single point somewhere along the length of the curve regardless of time.

Curve Radius

Objects scaled by the curve radius. See Curve Editing.

Follow Curve

If this option is not activated, the owner's rotation is not modified by the curve; otherwise, it is affected depending on the Forward and Up Axes.

Animate Path

Adds an F-Curve with options for the start and end frame. ToDo: from above.

Influence

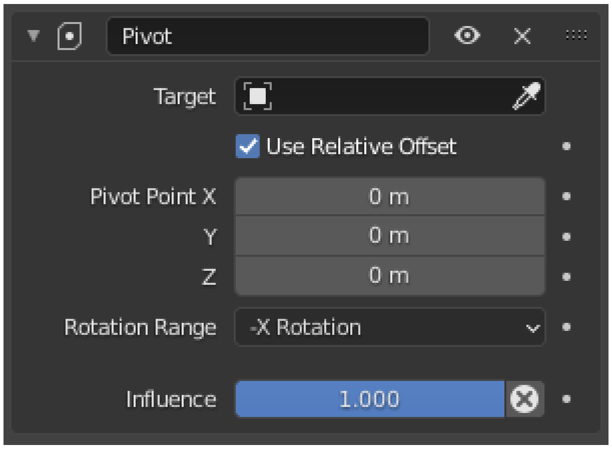
Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Pivot Constraint

The *Pivot* constraint allows the owner to rotate around a target object. It was originally intended for pivot joints found in humans e.g. fingers, feet, elbows, etc.

Options



Pivot

Target

☑ Use Relative Offset

Pivot Point X: 0 m

Y: 0 m

Z: 0 m

Rotation Range: -X Rotation

Influence: 1.000

Pivot panel.

Target

Data ID for the selection of the object to be used as a pivot point. See common constraint properties for more information.

Use Relative Offset

Offset will be an absolute point in space instead of relative to the target.

Pivot Point X, Y, Z

Offset of pivot from target.

Rotation Range

Rotation range on which pivoting should occur.

Always

Use the pivot point in every rotation.

-X/-Y/-Z/X/Y/Z Rotation

Use the pivot point in the corresponding direction around the corresponding axis.

Influence

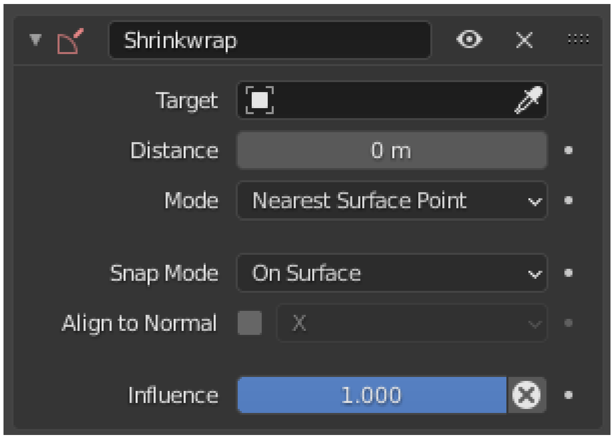
Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Example

Shrinkwrap Constraint

The *Shrinkwrap* constraint is the “object counterpart” of the Shrinkwrap Modifier. It moves the owner origin and therefore the owner object’s location to the surface of its target. This implies that the target *must* have a surface; thus, you can only use meshes as targets.

Options



Target
Distance: 0 m
Mode: Nearest Surface Point
Snap Mode: On Surface
Align to Normal: ☐ X
Influence: 1.000

Shrinkwrap panel.

Target

Data ID used to select the constraint’s target, which *must* be a mesh object, and is not functional (red state) when it has none. See common constraint properties for more information.

Distance

This number field controls the offset of the owner from the shrunk computed position on the target’s surface.

Influence

Controls the percentage of affect the constraint has on the object. See common constraint properties for more information.

Mode

This selector allows you to select which method to use to compute the point on the target's surface to which to move the owner's origin. You have these options:

Nearest Surface Point

The chosen target's surface's point will be the nearest one to the original owner's location. This is the default and most commonly useful option.

Projection

The target's surface point is determined by projecting the owner's origin along a given axis.

Project Axis

> This axis is controlled by the radio buttons that show up when you select this type. This mean the projection axis can only be aligned with one of the three axes, or their opposites. When the projection of the owner's origin along the selected direction does not hit the target's surface, the owner's location is left unchanged.
>
> +X, +Y, +Z, -X, -Y, -Z

Space

> Coordinate space in which the axis direction is specified.

Distance

> Distance cutoff after which projection is assumed to have failed, leaving the location unchanged.

Project Opposite

> In addition to the selected projection axis, project in the opposite direction and choose the closest hit.

Face Cull

This radio button allows you to prevent any projection over the “front side” (respectively the “back side”) of the target’s faces. The “side” of a face is determined by its normal (front being the side “from where” the normal “originates”).

Off, Front, Back

Invert Cull
When used with *Project Opposite* and *Face Culling*, it inverts the *Front* or *Back* cull choice for the opposite direction.

Nearest Vertex

This method is very similar to the *Nearest Surface Point* one, except that the owner’s possible shrink locations are limited to the target’s vertices.

This method doesn’t support the *Snap Mode* setting described below.

Target Normal Projection

This method is similar to *Nearest Surface Point*, but produces a much smoother projection in return for being significantly slower.

Instead of finding the closest point, it searches for the nearest point that has its interpolated smooth normal pointing towards or away from the original owner position. Non-manifold boundary edges are specially handled as infinitely thin cylinders that emit normals in all perpendicular directions; ignores flat shading.

Snap Mode

Most Shrinkwrap types support an additional setting to control how the owner is moved to the target point selected by the methods described above. Some of the choices only differ if *Distance* is not zero.

On Surface

The owner location is always changed. The offset is applied along the projection line connecting the original owner location and selected target point towards the original position.

Outside Surface
Like *On Surface*, but the offset is always applied towards the outside of the target.

Above Surface
Like *On Surface*, but the offset is applied along the smooth normal of the target.

Inside
The owner is not moved if it is already inside the target. Offset shrinks the allowed volume towards the inside along the projection line.

Outside
The owner is not moved if it is already outside the target. Offset expands the exclusion volume towards the outside along the projection line.

The *Inside* and *Outside* options can be used for very crude collision detection. The inside vs outside determination is done based on the target normal and is not always stable near 90 degree and sharper angles in the target mesh.

Align To Normal

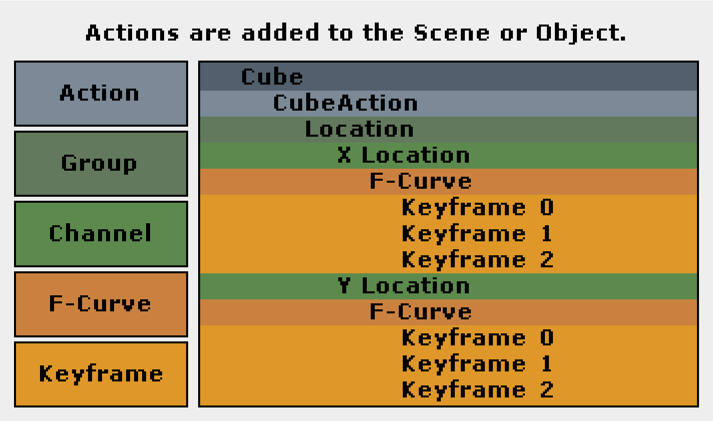
Whenever *Snap Mode* is available, it is also possible to align the specified local axis of the object to the smooth normal of the target at the selected point. The axis is selected via radio buttons.

The alignment is performed via smallest rotation, like in Damped Track constraint.

Example

Actions

When animating objects and properties in Blender, Actions record and contain the data. As everything else in Blender, Actions are data-blocks.



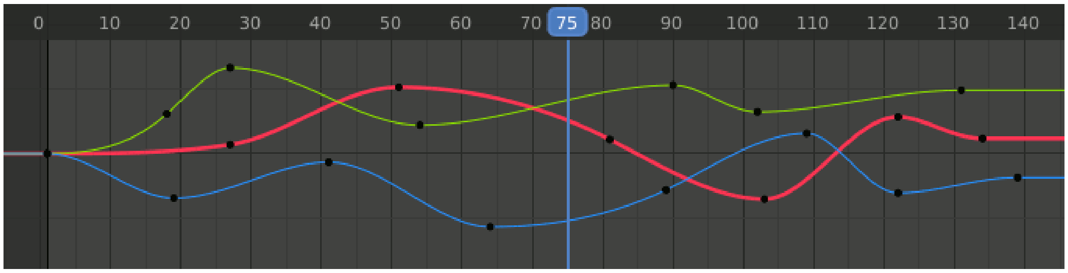
Actions are added to the Scene or Object.

Action
Group
Channel
F-Curve
Keyframe
Cube
CubeAction
Location
X Location
F-Curve
Keyframe 0
Keyframe 1
Keyframe 2
Y Location
F-Curve
Keyframe 0
Keyframe 1
Keyframe 2

Actions.

So when you animate an object by changing its location with keyframes, the animation is saved to the Action.

Each property has a channel which it is recorded to, for example, `Cube.location.x` is recorded to Channel X Location. The *X location* and *Y location* properties can be shared across multiple objects, if all objects have *X location* and *Y location* properties beneath them.



0
10
20
30
40
50
60
70
75
80
90
100
110
120
130
140

Graph Editor. Each channel has an F-Curve represented by the lines between the keyframes.

Actions

Record and contain animation data.

Groups

Are groups of channels.

Channels

Record properties.

F-Curves

F-Curves are used to interpolate the difference between the keyframes.

Keyframes

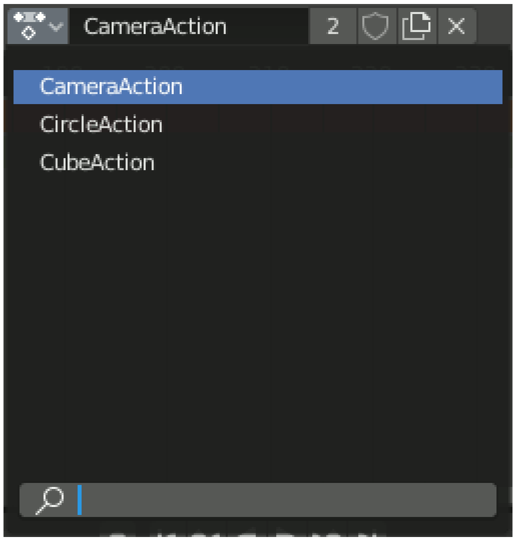
Keyframes are used to set the values of properties bound to a point in time.

Working with Actions

When you first animate an object by adding keyframes, Blender creates an *Action* to record the data.

Actions can be managed with the *Action data-block menu* in the Dope Sheet Action Editor header, or the Sidebar region of the NLA Editor.

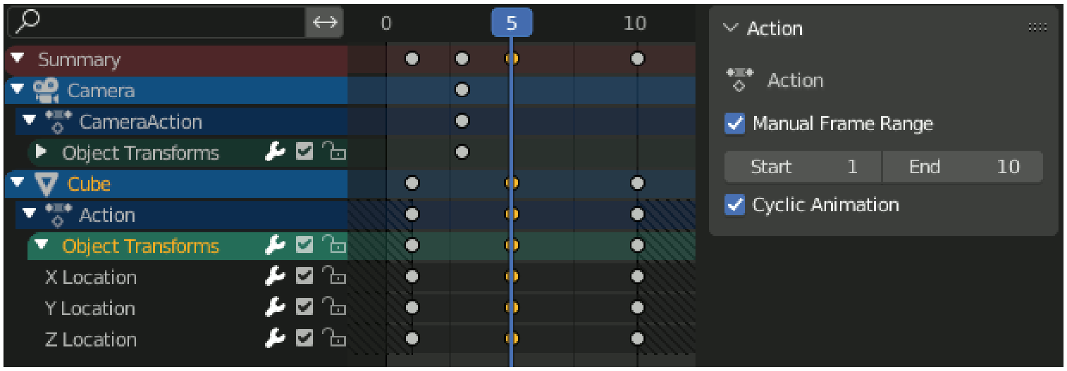
If you are making multiple actions for the same object, press the shield button for each action. This will give the actions a *Fake User* and will make Blender save the unlinked actions.



The Action data-block menu.

Objects can only use one *Action* at a time for editing. The NLA Editor is used to blend multiple actions together.

Properties



☑ Manual Frame Range

Start 1 End 10

☑ Cyclic Animation

Actions with and without a Manual Frame Range in Dope Sheet.

It is possible to manually specify the intended useful frame range of an action via a panel available in the [Dope Sheet]() or the [NLA Editor]() when a channel or NLA track is selected.

Manual Frame Range

Manually specify the intended playback frame range for the action (this range is used by some tools, but does not affect animation evaluation). The manual frame range feature can be toggled with the checkbox.

When the range is set, it is used instead of the actual range occupied by key frames when adding a new track based on the action to NLA. It can also be used by exporters to determine the range of frames to export.

The range is displayed in the background of the editor as diagonal hash fill, to distinguish it from the solid fill of the current playback range.

The frame values are most commonly expected to be integers, but can be fractional.

Cyclic Animation

Specifies that the action is intended to be cyclic over the specified range. The first and last frames of the range should represent the same pose of the cycle one loop apart, i.e. the range should include the duplicated initial key of the loop.

Note

This option signifies intent and does **not** make the action cycle on its own. However, if Cycle-Aware Keying is enabled, it will automatically enable cyclic extrapolation and set up the loop period for curves newly added to the action.

Custom Properties

Create and manage your own properties to store data in the action's data block. See the Custom Properties page for more information.

Drivers

- Introduction
  - Graph View
  - Driver Configuration
  - Notes on Scripted Expressions
- Usage
  - Add Driver
  - Edit Driver
  - Open Drivers Editor
  - Copy & Paste
  - Copy Driver to Selected
  - Copy As New Driver
  - Expression
  - Removing Drivers
- Drivers Panel
  - Driver Settings
  - Driver Variables
  - Expressions
- Workflow & Examples
  - Transform Driver
  - Scripted Expression - Orbit a Point
  - Custom Function - Square Value
  - View Layer Attribute Lookup
  - Shape Key Drivers
- Troubleshooting
  - Scripted Expression
  - Rotational Properties are Radians

Introduction

Drivers are a way to control values of properties by means of a function, or a mathematical expression.

Effectively, drivers consist of:

- A **driver configuration** that specifies zero, one, or more input values using other properties or object transformation channels, and combines them using a predefined mathematical function or a custom Python expression.
- An **animation** F-Curve that maps the output of the driver configuration to the final value to apply to the driven property.

As an example, the rotation of Object 1 can be controlled by the scale of Object 2. It is then said that the scale of Object 2 *drives* the rotation of Object 1.

Not only can drivers directly set the value of a property to the value of a different one, they can also combine multiple values using a fixed function or a Python expression and further modulate it with a manually defined curve and/or a modifier stack.

Drivers are an extremely powerful tool for building rigs and are typically used to drive bone transforms and the influence of shape keys, action constraints and modifiers, often using custom properties as inputs.

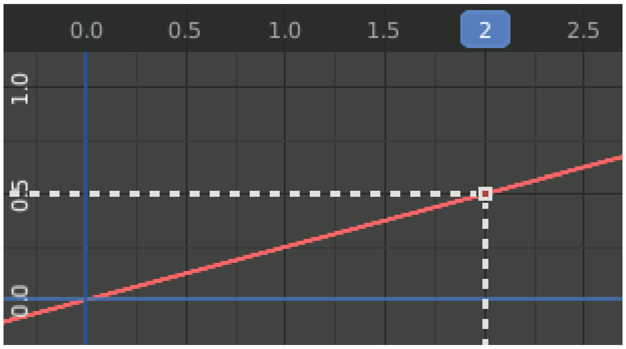
Graph View

The main area of the Drivers editor shows an F-Curve that represents the driver function.

The **X axis** maps to the output value of the driver configuration. The units depend on the setup.

The **Y axis** shows the value applied to the target property. The units depend on the property.

In the example image, if the driver value is 2.0 the property value will be 0.5.



Driver curve in the Drivers editor.

The default F-Curve is an identity map, i.e. the value produced by the driver configuration is applied to the driven property unchanged. If the driver output value is 2.0, the property will be 2.0.

The driver function can be defined artistically with Bézier curve handles or mathematically with trigonometric functions or polynomial expressions such as \(y = a + bx\). Furthermore, the function can also be procedurally modulated with noise or cyclic repetitions. See Modifiers for more details.

Driver Configuration

The Drivers panel shows the setup for a driver.

A driver can have zero, one, or more **variables**. Variables specify which properties, object transformation channels, or relative distances between objects, are used as inputs by the driver.

The driver **type** determines how the variables are used. The type can be:

- a built-in function: for example, the sum of the variables' values, or
- a scripted expression: an arbitrary Python expression that refers to the variables by their names.

This driver configuration outputs a single value which changes when the variables change. This value is then evaluated through the driver function curve to produce the result to be applied to the driven property.

Notes on Scripted Expressions

When a driver uses a *Scripted Expression*, Blender can evaluate it without using the fully featured Python interpreter if it is simple enough. This means that drivers are fast to evaluate with simple divisions, additions and other “simple” expressions. The built-in functions are always evaluated natively.

See Simple Expressions for a comprehensive list of expressions that can be evaluated natively.

When the expression is not simple, it will be evaluated using Python. As a consequence, the driver will be slower and there is a security risk if the author of the Python code is unknown. This is an important thing to take into consideration for heavy scenes and when sharing files with other people. See also: Auto run.

Usage

Drivers can be added to properties via their context menu, a shortcut, copy-pasted, or by typing an expression directly into the property's value.

After adding drivers, they are usually modified in the Drivers editor, or via a simplified *Edit Driver* popover invoked from the property context menu.

Add Driver

Reference

Menu::	*Context menu* ‣ *Add Driver*
Shortcut::	`Ctrl-D`

The usual way to add a driver to a property is to `RMB` click a property, then choose *Add Driver* in the context menu. Drivers can also be added by pressing `Ctrl-D` with the mouse over the property.

This operation adds a driver with a single variable (which needs to be filled in), and displays the *Edit Driver* popover.

Edit Driver

Reference

Menu::	*Context menu* ‣ *Edit Driver*

Displays a popover window that allows editing the custom expression and input variables of the driver without opening the full *Drivers Editor*.

Many drivers don't use their F-Curve component, so this reduced interface is sufficient.

Open Drivers Editor

Reference

Menu:: *Context menu ‣ Open Drivers Editor*

Opens a new window with the *Drivers Editor* and selects the driver associated with the property.

Copy & Paste

Reference

Menu:: *Context menu ‣ Copy Driver*
Menu:: *Context menu ‣ Paste Driver*

Drivers can be copied and pasted via the context menu. When adding drivers with the same settings, this can save time modifying settings.

Copy Driver to Selected

Reference

Menu:: *Context menu ‣ Copy Drivers to Selected*
Menu:: *Context menu ‣ Copy Driver to Selected*
Menu:: *Context menu ‣ Copy All Drivers to Selected*

Copy the property's driver from the active item to the same property of all selected items, if the same property exists.

Copy As New Driver

Reference

Menu::	*Context menu ‣ Copy As New Driver*

A driver that sets the property value to the value of a different property can be quickly created by using the *Copy As New Driver* context menu option of the input property, and then pasting the result onto the output property via *Paste Driver*.

It is also possible to add the new driver variable to an existing driver using the Paste Driver Variables button in the editor panel.

Expression

This is a quick way to add drivers with a scripted expression. First click the property you want to add a driver to, then type a hash # and a scripted expression.

Some examples:

- `#frame`
- `#frame / 20.0`
- `#sin(frame)`
- `#cos(frame)`

Removing Drivers

Reference

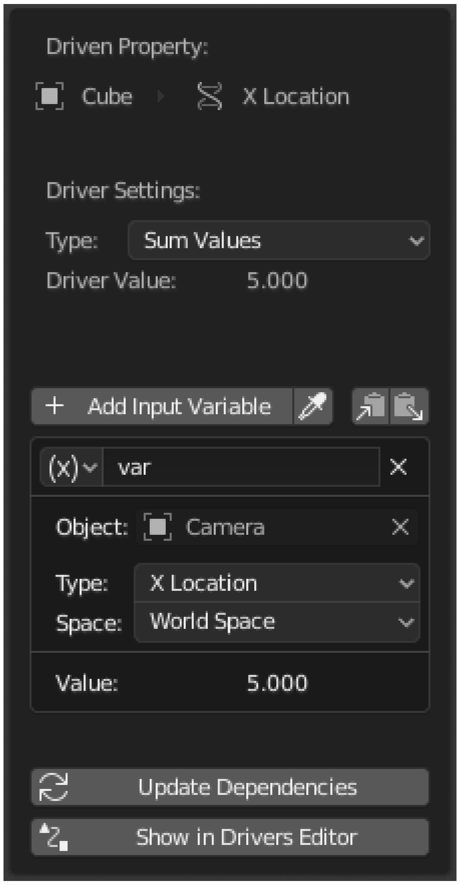
Menu::	*Context menu ‣ Delete Driver(s)*
Menu::	*Context menu ‣ Delete Single Driver*
Shortcut::	`Ctrl-Alt-D`

Removes driver(s) associated with the property, either for the single selected property or sub-channel, or all components of a vector.

Drivers Panel

Reference

Editor::


Add Input Variable
Update Dependencies
Show in Drivers Editor

Driven Property: Cube › X Location

Driver Settings:

Type: Sum Values

Driver Value: 5.000

(x) var

Object: Camera

Type: X Location

Space: World Space

Value: 5.000

Edit Driver popover.

Graph editor

Mode:: Drivers

Panel:: *Sidebar region ‣ Drivers*

Shortcut:: `N`

Reference

Menu:: *Context menu ‣ Edit Driver*

Shortcut:: `Ctrl-D`

This panel is visible in Sidebar of the Drivers Editor or as a popover when adding a driver to a property.

It shows the property that is being driven, followed by a series of settings that determine how the driver works.

Driver Settings

Type

There are two categories of drivers:

- **Built-in functions** (*Average*, *Sum*, *Min* and *Max*)

  The driven property will have the value of the average, sum, lowest or highest (respectively) of the values of the referenced *Driver Variables*. If there is only one driver variable, these functions will yield the same result.

- **Custom** (*Scripted Expression*).

  An arbitrary Python expression that can refer to the *Driver Variables* by name. See Expressions.

Driver Value

The current result of the driver setup. Useful for debug purposes.

Variables

See Driver Variables.

Update Dependencies

Forces an update for the Driver Value dependencies.

Show in Drivers Editor

Opens the fully featured Drivers Editor. This button only appears in the popover version of the Drivers panel.

Driver Variables

Variables are references to properties, transformation channels, or the result of a comparison between transformations of two objects.

Drivers should access object data via *Driver Variables*, rather than direct references in the Python expression, in order for dependencies to be correctly tracked.

Add Input Variable
    Adds a new Driver Variable.

Add Input Variable

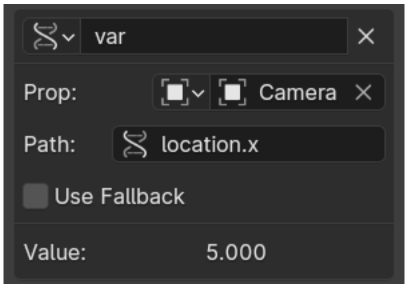
Add, Copy, Paste buttons.

Copy/Paste Variables
    Copies the current variable list so it can be pasted into another driver's variable list.

Name
    Name for use in scripted expressions. The name must start with a letter, and only contain letters, digits, or underscores.

Variable Type
    The type of variable to use.

Single Property
    Retrieves the value of an RNA property, specified by a data-block reference and a path string.



var

Prop: Camera

Path: location.x

☐ Use Fallback

Value: 5.000

In case of transform properties, this will return the exact value of the UI property, while Transform Channel will take parenting and/or constraints into account as needed.

See also Custom Properties.

ID Type
The ID-block type. For example: Key, Image, Object, Material.

ID
The ID of the ID-block type. For example: “Material.001”.

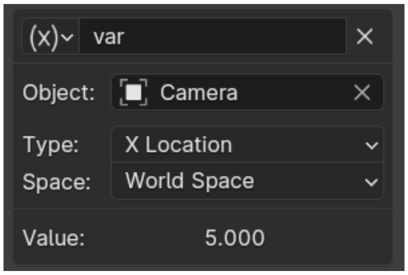
RNA Path
The RNA name of the property, based on a subset of Python attribute access syntax. For example: `location.x` or `location[0]` for the X location animation channel value (before parenting or constraints), or `["prop_name"]` for a custom property.

Fallback
If enabled, allows specifying a fallback value to use as the variable value if the RNA Path cannot be resolved, instead of causing a driver evaluation failure. For more info see Context Property below.

Tip

The easiest way to create a variable of this type is to use the Copy As New Driver context menu option of the input property, and paste the result into the driver via Paste Driver Variables.

Transform Channel
Retrieves the value of a Transform channel from an object or bone.


(x) var

Object: Camera

Type: X Location

Space: World Space

Value: 5.000

ID
ID of the object. For example: Cube, Armature, Camera.

Bone

For armatures, the name of the Armature bone. For example: “Bone”, “Bone.002”, “Arm.r”.

Type

For example, X Location, X Rotation, X Scale.

The *Average Scale* option retrieves the combined scale value, computed as the cubic root of the total change in volume. Unlike *X/Y/Z Scale*, this value can be negative if the object is flipped by negative scaling.

Mode (Rotation)

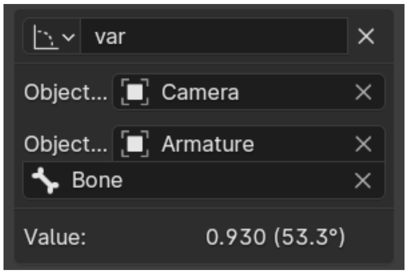
For rotation channels, specifies the type of rotation data to use, including different explicit [Euler]() orders. Defaults to using the Euler order of the target. See [Rotation Channel Modes]().

Space

World Space, Transform Space, Local Space.

Rotational Difference

Provides the value of the rotational difference between two objects or bones, in radians.



var

Object... Camera

Object... Armature

Bone

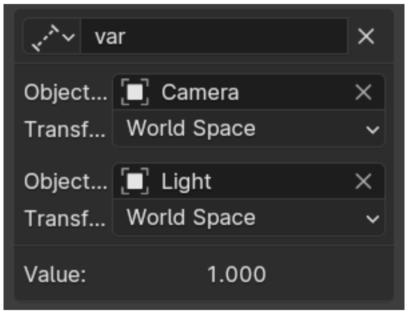
Value: 0.930 (53.3°)

Bone

For armatures, the name of the Armature bone. For example: “Bone”, “Bone.002”, “Arm.r”.

Distance

Provides the value of the distance between two objects or bones.



var

Object... Camera

Transf... World Space

Object... Light

Transf... World Space

Value: 1.000

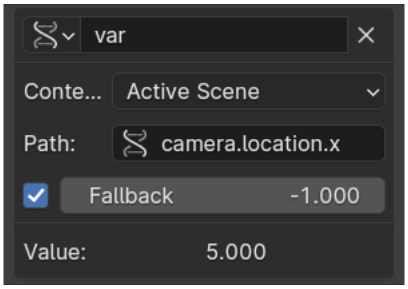
Bone

For armatures, the name of the Armature bone. For example: “Bone”, “Bone.002”, “Arm.r”.

Space

World Space, Transform Space, Local Space.

Context Property


var

Conte... Active Scene

Path: camera.location.x

☑ Fallback -1.000

Value: 5.000

Provides the value of a property that is implicitly referring to either a scene or a view layer of the currently evaluating animation system. This is a weak reference which does not lead to the scene or view layer referenced from the driver to be linked when linking animation data.

An example when such properties comes in play is referring to a transformation of the active camera. It is possible to set up a driver in a character file, and make the driver use the set camera when the character is linked into a set.

Context

Active Scene, Active View Layer.

RNA Path

The RNA name of the property, based on a subset of Python attribute access syntax. For example: `camera.location.x` or `camera.location[0]` for the camera X location animation channel value (before parenting or constraints), or `["prop_name"]` for a custom property.

Fallback

If enabled, allows specifying a fallback value to use as the variable value if the RNA Path cannot be resolved, instead of causing a driver evaluation failure.

This feature can be very useful for making drivers more robust when implementing scene-global options using custom properties. When the object is linked into a different scene, these custom properties may not exist there, and the fallbacks can be used to provide sensible default values.

Fallbacks can also be used to emulate the lookup behavior of the View Layer mode of the material Attribute Node.

Tip

Although the values of the x/y/z animation channels for the camera location can be accessed via `camera.location[0/1/2]`, retrieving its world space location and orientation after parenting and constraints currently requires using `camera.matrix_world`. This property can be understood easily by viewing the matrix as an array of four vectors in *World* space:

- `matrix_world[0][0/1/2]` is the *Screen Right* direction vector (camera local X).
- `matrix_world[1][0/1/2]` is the *Screen Up* direction vector (camera local Y).
- `matrix_world[2][0/1/2]` is the **opposite** of the direction the camera is pointing.
- `matrix_world[3][0/1/2]` is the *location* of the camera.

Value

Shows the value of the variable.

Rotation Channel Modes

Rotation Transform Channels support a number of operation modes, including:

Auto Euler

Uses the Euler order of the target to decompose rotation into channels.

XYZ Euler, …

Explicitly specifies the Euler rotation order to use.

Quaternion

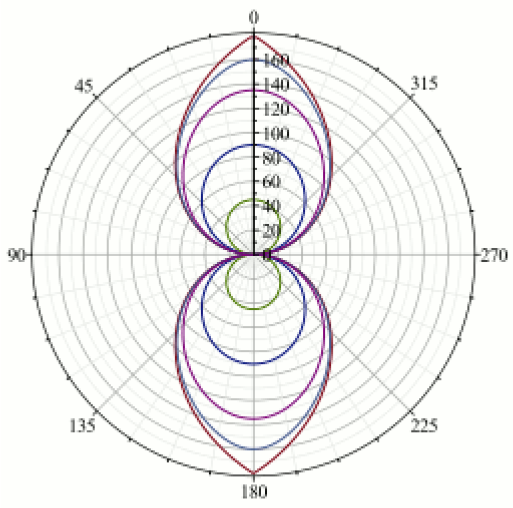
Provides the Quaternion representation of the rotation.

Swing and X/Y/Z Twist

Decomposes the rotation into two parts: a Swing rotation that aims the specified axis in its final direction, followed by a Twist rotation around that axis. This is often necessary for driving corrective Shape Keys and bones for organic joint rotation.

This decomposition is often produced in rigs by using a helper bone with a Damped Track Constraint to extract the swing part, and its child with Copy Transforms to extract the twist component.

The channels values for *Swing and Y Twist* are:



Falloff curves for weighted angles.

Y Rotation

True angle of the twist rotation.

W Rotation

True angle of the swing rotation, independent of its direction.

X Rotation, Z Rotation

Weighted angles that represent the amount of swing around the X/Z axis.

The magnitude of the angle equals *W Rotation* when the rotation is purely around that axis, and fades out to zero as the direction changes toward the other axis, following the falloff curves from the graph on the right.

Mathematically, the swing angles are computed from quaternion components, using \(2 \arccos(w)\) for *W* and \(2 \arcsin(x)\) etc. for the others. The component of the swing rotation that corresponds to the twist axis is always 0, and is replaced by the twist angle.

Expressions

Expression

A text field where you can enter an arbitrary Python expression that refers to *Driver Variables* by their names.

The expression has access to a set of standard constants and math functions from `math`, `bl_math` and other modules, provided in the *Driver Namespace*. For an example of adding a custom function to the namespace, see the driver namespace example.

For performance reasons it is best to use the Simple Expressions subset as much as possible.

Use Self

If this option is enabled, the variable `self` can be used for drivers to reference their own data. Useful for objects and bones to avoid having creating a *Driver Variable* pointing to itself.

Example: `self.location.x` applied to the Y rotation property of the same object will make the object tumble when moving.

Note that dependencies for properties accessed via `self` may not be fully tracked.

Simple Expressions

Blender can evaluate a useful subset of Python driver expressions directly, which significantly improves performance, especially on multi-core systems. To take advantage of this, the driver expression must only use the following features:

Variable Names

Use only ASCII characters.

Literals

Floating-point and decimal integer.

Globals

`frame`

Constants

`pi`, `True`, `False`

Operators

`+`, `-`, `*`, `/`, `==`, `!=`, `<`, `<=`, `>`, `>=`, `and`, `or`, `not`, conditional operator/ ternary if

Standard Functions

`min`, `max`, `radians`, `degrees`, `abs`, `fabs`, `floor`, `ceil`, `trunc`, `round`, `int`, `sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `atan2`, `exp`, `log`, `sqrt`, `pow`, `fmod`

Blender Provided Functions

`lerp`, `clamp`, `smoothstep`

Simple expressions are evaluated even when Python script execution is disabled.

When an expression outside of this subset is used, Blender displays a “Slow Python expression” warning. However, as long as the majority of drivers use simple expressions, using a complex expression in select few is OK.

See also

- Extending Blender with Python.
- Python and its documentation.
- functions.wolfram.com.

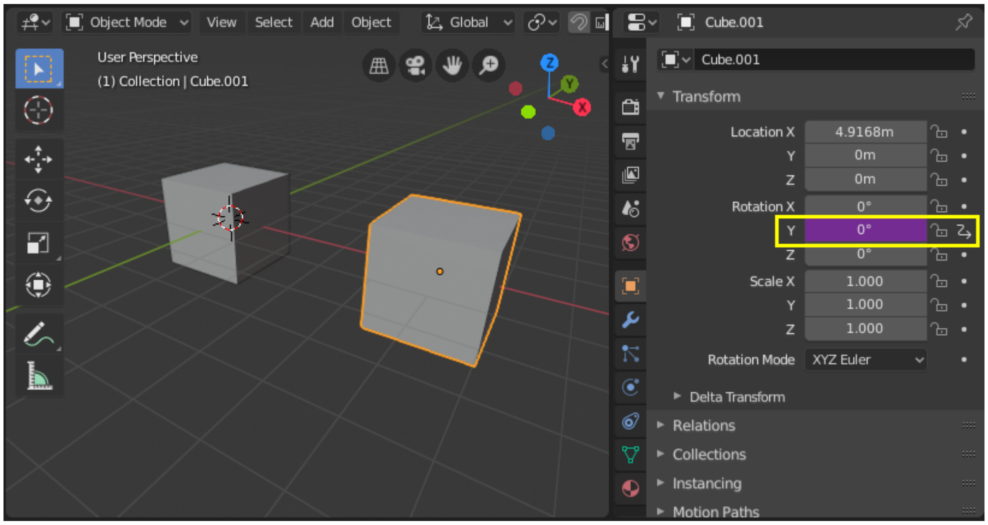
Workflow & Examples

Simple Drivers can be configured from the pop-over that appears when adding a new Driver. When adding multiple Drivers or for more advanced configurations, it is useful to have open the Drivers Editor.

Transform Driver

Control a property with an object's transform. In this example, the Y rotation of Object 2 will be driven by the X position of Object 1. Starting from a simple setup with two objects:

1. Add a Driver to the *Rotation Y* property of the second object via the context menu or with `Ctrl-D`.



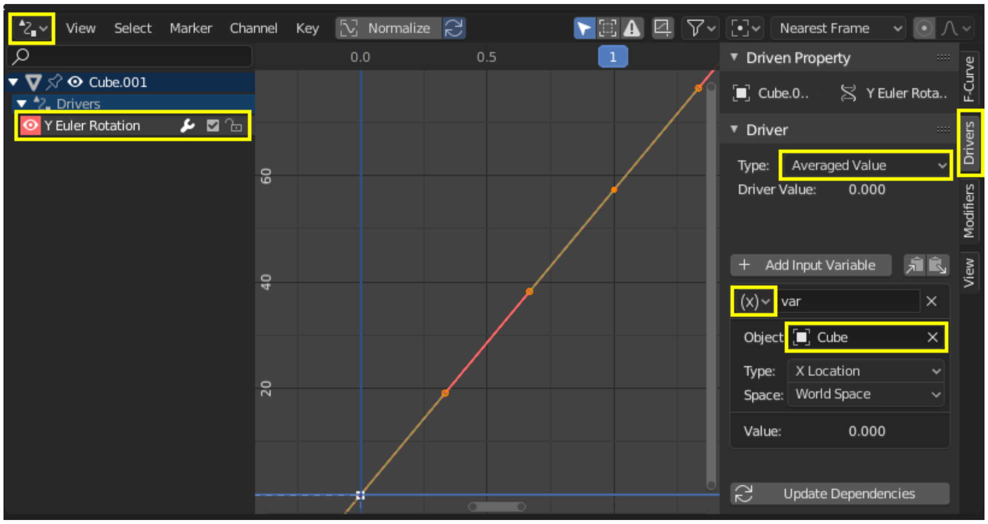
Object Mode
View
Select
Add
Object
Global
Cube.001
User Perspective
(1) Collection | Cube.001
Cube.001
Transform
Delta Transform
Relations
Collections
Instancing
Motion Paths

Location X 4.9168m
Y 0m
Z 0m
Rotation X 0°
Y 0°
Z 0°
Scale X 1.000
Y 1.000
Z 1.000
Rotation Mode XYZ Euler

2. Open the *Drivers Editor* and select the *Y Euler Rotation* property in the channels region.

3. Open the Sidebar region and select the *Drivers* tab.

4. Configure the driver to be the *Averaged Value* of a *Transform Channel* of the first object.

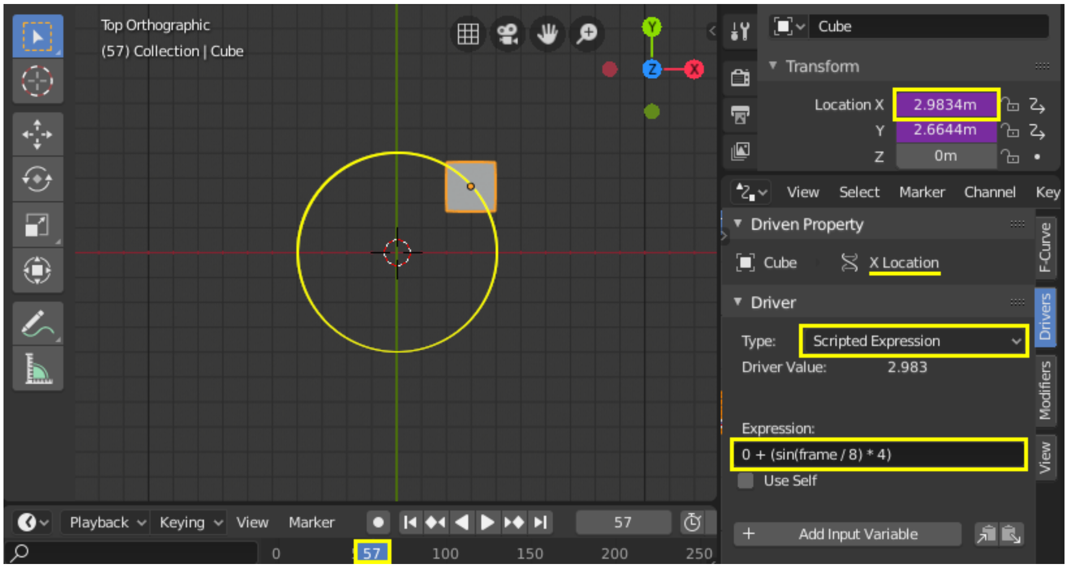

View
Select
Marker
Channel
Key
Normalize
Nearest Frame
Cube.001
Drivers
Y Euler Rotation
0.0
0.5
1
60
40
20
Driven Property
Cube.0..
Y Euler Rota..
Driver
Add Input Variable
var
Update Dependencies
F-Curve
Drivers
Modifiers
View

Type: Averaged Value
Driver Value: 0.000
Object: Cube
Type: X Location
Space: World Space
Value: 0.000

5. Experiment with moving the first object and notice how it affects the Y rotation of the second object.

Scripted Expression - Orbit a Point

Orbit an object’s position around a point with a custom *Scripted Expression*. The object’s position will change when scrubbing the timeline. Using trigonometry, circular motion can be defined in 2D using the sine and cosine functions. (See Unit Circle.) In this example, the current frame is used as the variable that induces the motion. `frame` is a Simple Expression that corresponds to `bpy.context.scene.frame_current`.


Top Orthographic
(57) Collection | Cube
Y
Z
X
Cube
Transform
View
Select
Marker
Channel
Key
Driven Property
Cube
X Location
Driver
Use Self
Add Input Variable
Playback
Keying
View
Marker
57
0
57
100
150
200
250
F-Curve
Drivers
Modifiers
View

Location X 2.9834m
Y 2.6644m
Z 0m
Type: Scripted Expression
Driver Value: 2.983
Expression: 0 + (sin(frame / 8) * 4)

1. Add a driver to the X Location property.
    1. Set the *Driver Type* to *Scripted Expression*.
    2. Add the expression `0 + (sin(frame / 8) * 4)`, where:
        - `frame/8` : is the current frame of the animation, divided by 8 to slow the orbit down.
        - `(sin( )*4)` : multiplies the result of `sin(frame/8)` by 4 for a bigger circle.
        - `0 +` : is used to control the offset to the orbit center point.
2. Add a driver to the Y Location property with the expression `0 + (cos(frame / 8) * 4)`.
3. Scrub the timeline to see the effect. Experiment with the variables to control the size and center of the orbit.

Custom Function - Square Value

Create a custom function to get the square of a value (i.e. $value^2$). Adding the function to the *Driver Namespace* allows it to be used from driver expressions. The *Driver Namespace* has a list of built-in functions for use in driver expressions, as well as constants such as π and e. These can be inspected via the Python Console:

```
>>> bpy.app.driver_namespace[' <tab>
                              acos']
                              acosh']
                              asin']
                              asinh']
                              atan']
                              ...
```

To add a new function to the *Driver Namespace*, the function itself needs to be implemented and then added to the `bpy.app.driver_namespace`.

1. Add the following to the Text Editor inside Blender and press *Run Script*.

    ```
    import bpy

    def square(val):
    ```

```
    """Returns the square of the given value"""
    return val * val

# Add function to driver_namespace.
bpy.app.driver_namespace['square'] = square
```

2. Add a driver with a *Scripted Expression* such as `square(frame)`.

3. Observe the effect when scrubbing the timeline.

There are more custom function examples available in Blender’s Text Editor *Templates* ‣ *Python* ‣ *Driver Functions*. Since Simple Expressions cannot access custom functions, using them only makes sense for complex computations.

Warning

Trying to replace built-in entries of the driver namespace may result in undefined behavior.

View Layer Attribute Lookup

The material Attribute Node in the View Layer mode automatically searches for the attribute property in multiple locations. This, for example, can allow setting a certain value of the custom attribute at the Scene or World level, and then overriding it differently for one View Layer.

Context Properties of drivers don’t implement this behavior, so if necessary it has to be manually emulated via fallback values and a conditional expression (conditions are Simple Expressions).

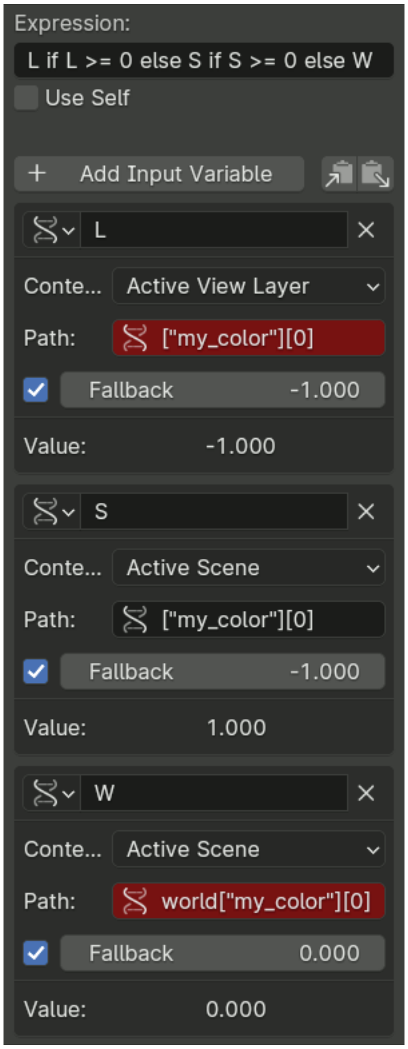
For an attribute named `attr`, the node tries the following six RNA path lookups in order:

- `["attr"]` in the active View Layer (custom property).
- `attr` in the active View Layer (built-in property).
- `["attr"]` in the active Scene.

- `attr` in the active Scene.
- `world["attr"]` in the active Scene.
- `world.attr` in the active Scene.

Depending on the specific property it may be sufficient to check only a subset of these locations. For example, the image on the right shows how to access an attribute that is known to definitely be a custom property with a color value.

Driver variables accessing locations that are not final in the lookup chain should use fallback values that are invalid for the attribute (e.g. negative color values), which can then be checked by the conditional expression. The final variable should fallback to a valid default value to be used when the property is not set at all.



Expression: L if L >= 0 else S if S >= 0 else W
☐ Use Self

L
Conte... Active View Layer
Path: ["my_color"][0]
☑ Fallback -1.000
Value: -1.000

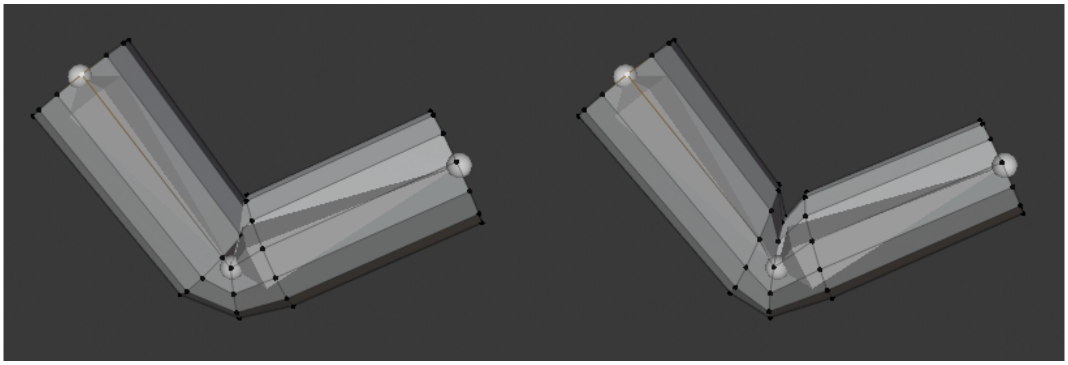
S
Conte... Active Scene
Path: ["my_color"][0]
☑ Fallback -1.000
Value: 1.000

W
Conte... Active Scene
Path: world["my_color"][0]
☑ Fallback 0.000
Value: 0.000

Shape Key Drivers

Improved Mesh Deformation

Fix intersection problems that happen when using armatures and weight painting, especially at joints. Shape keys can also be used to tweak and refine a rig, for example to suggest muscle formations. In this example, a shape key is used to improve the deformation at the elbow of a rudimentary arm.



Left: Skeletal mesh deformation without correction. Right: Corrective shape key applied

Setup

1. Add a mesh (in this example, a cylinder with loop cuts).
2. Add an armature with a chain of bones.
3. Skin the mesh to the armature using weight painting.

(Note: to parent the mesh to the armature: select the mesh first, then the armature and use `Ctrl-P` to parent with auto weights.)

Experiment with posing the armature and observe the deformation at the joint. To fix intersection problems or angles that look unsatisfactory, you can associate a Shape Key with a pose.

Shape Key

1. Pose the armature such that the problems are visible. Be sure to cover the extreme poses that you want to support for the rig.
2. With the mesh selected, add a new *Shape Key* in addition to the *Basis* key. *Properties* ‣ *Mesh tab* ‣ *Shape Keys*
3. In order to author the shape key on top of the armature deformation, enable both *Edit Mode Display* and *Cage Editing* in the Armature modifier. *Properties* ‣ *Modifiers tab* ‣ *Armature Modifier* ‣ *Header*
4. Enter Edit Mode and select the new shape key in the properties panel. Adjust the vertices as desired. Select the *Basis* key to toggle between the original mesh and your edits. (Note: be careful to apply edits only to your shape and not to the original mesh or other existing keys.)

Once you are satisfied with how the deformation looks for the problematic pose, you'll need to configure a driver to activate the shape smoothly when entering that position.

Driver

1. Add a driver to the *Value* of the shape key you've created.
2. Open the Drivers Editor and select the driver.

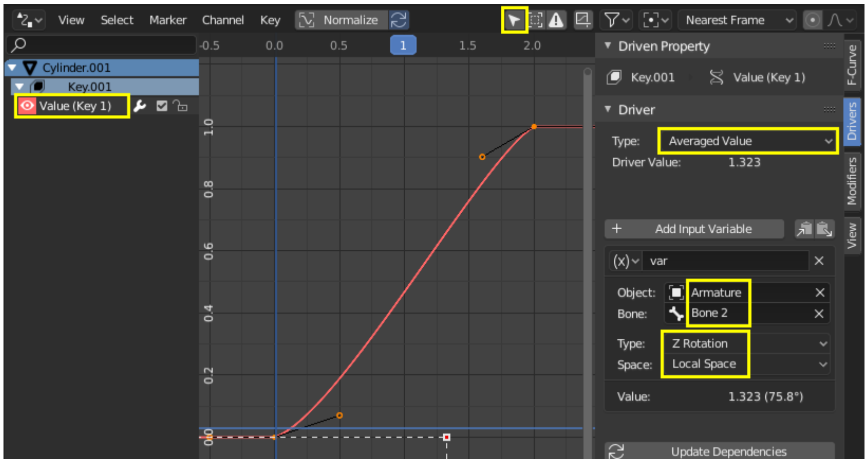
Method 1 – Direct mapping to a bone rotation value

A simple way to configure the driver is with a direct correspondence of the value of a bone's rotation channel to the shape key activation *Value*. This method has the disadvantage of relying on a single channel of a bone's rotation which might be insufficient to precisely express the condition under which the shape key should be activated.

1. In the Drivers tab, select the *Averaged Value* of the rotation of the bone you are posing.

   Understand the rotation axis that you are interested in by enabling axes display in the armature or by observing the bone's transform values in the Properties.

   Select the rotation channel and set it to local, meaning, the bone's rotation value relative to its parent bone.



View
Select
Marker
Channel
Key
Normalize
Nearest Frame
-0.5
0.0
0.5
1
1.5
2.0
1.0
0.8
0.6
0.4
0.2
0.0
Cylinder.001
Key.001
Value (Key 1)
F-Curve
Drivers
Modifiers
View
Add Input Variable
Update Dependencies

Driven Property: Key.001 Value (Key 1)

Driver

Type: Averaged Value

Driver Value: 1.323

var

Object: Armature

Bone: Bone 2

Type: Z Rotation

Space: Local Space

Value: 1.323 (75.8°)

2. Manually set points in the driver curve by selecting a handle and dragging it or inserting values in the *F-Curve* tab. The Y

axis represents the shape key *Value*, which should go from 0.0 to 1.0. The X axis is usually the frame, but for this driver it represents the rotation value in radians. You can have more than two points in the curve and tweak the transitions with the handles in the curve view (`G` to move).

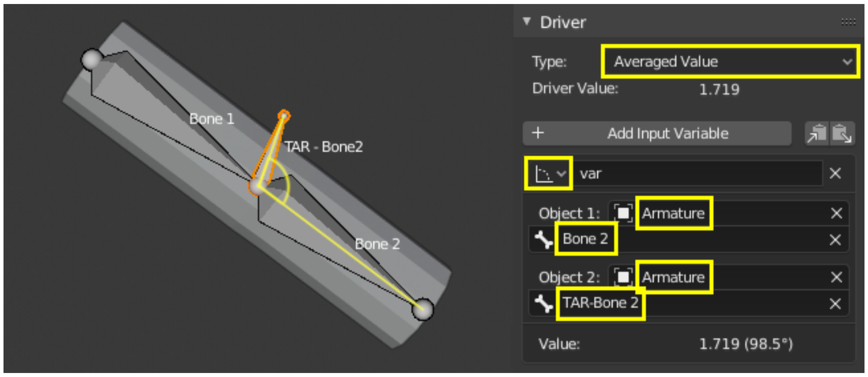
3. To verify that the driver behaves correctly, deselect the option to only show drivers for selected objects. This way, you can pose the armature and keep an eye on the driver.

Method 2 – Rotational difference to a target bone

This method requires an additional *target* or *corrective* bone, but it better expresses the spatial condition in 3D space of the bone that is causing the problem.

1. In armature Edit Mode, add a new bone extruded from Bone 1, in the position at which Bone 2 should have the shape key active. This type of bones usually follow a naming convention such as “TAR-” (target) or “COR-” (corrective).

2. In the Drivers tab, select the *Averaged Value* of the rotational difference between the bone you are rotating and the target bone. A rotational difference is the minimum angle between two objects in World Space. It is therefore important that the bones have the same root, so that the only thing affecting the angle between the bones is the rotation of one of them. When the deformation bone (Bone 2) reaches the target rotation (TAR-Bone 2) the rotational difference will be 0°.



Type: Averaged Value
Driver Value: 1.719
var
Object 1: Armature
Bone 2
Object 2: Armature
TAR-Bone 2
Value: 1.719 (98.5°)

3. Manually adjust the driver curve handles so that the shape key *Value* (Y axis) is 1.0 when the rotational difference (X axis) is 0°. The *Value* should be 0.0 when the arm is extended, at which point the rotational difference should be around 90° or more (in radians).

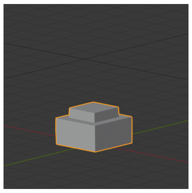
4. See the steps in Method 1 on how to adjust the curve handles and confirm that the functionality is working. Pose the armature to verify that the ranges are correct.

Chained Relative Shape Keys

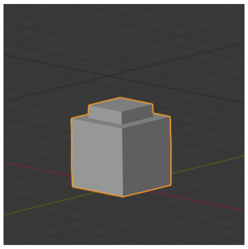
Activate different shape keys in succession. In this example, moving a single bone will activate first *Key 1* and then *Key 2*. See also relative shape keys mix additively.

Shape Keys

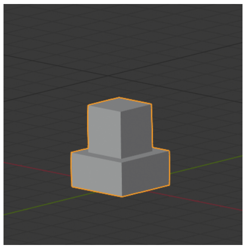
Add two shape keys to a mesh, besides the *Basis*.



Basis.



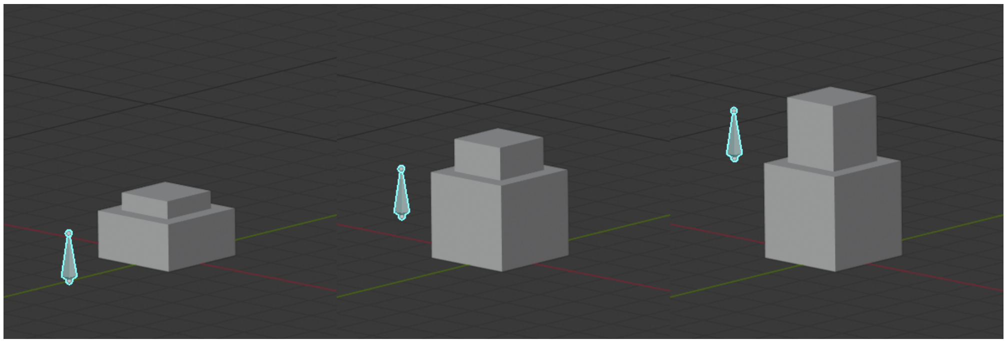
Key 1: top faces moved up by 1 m.



Key 2: inner top moved up by 1 m.

Drivers

Add an armature with a single bone to control the shape keys. The goal is to activate the keys in succession as this bone moves up.



As shown in the picture above, when the bone is halfway up, both *Key 1* and *Key 2* have an influence. It is a matter of preference if *Key 1* should be at its maximum *Value* before *Key 2* starts to become active, or how much they should overlap. This example shows a seamless blend.

For a seamless blend where there is overlap, *Key 1* should have a *Value* of 0.0 when the bone is at the bottom and increase linearly to 1.0 until the bone is past the midpoint height. *Key 2* should have a value of 0.0 before the midpoint height and then increase at the same rate than *Key 1* until reaching *Value* 1.0 when the bone is at maximum height.

1. Add a driver to the *Value* of *Key 1* and *Key 2*. In the *Drivers* tab, configure both drivers to be the *Averaged Value* of a variable with the bone's Z location.

2. Determine the range of the bone's motion in the World Z axis by moving it up so that it is aligned with the top of the mesh when both keys are active. Here we will use [0.0, 2.5].

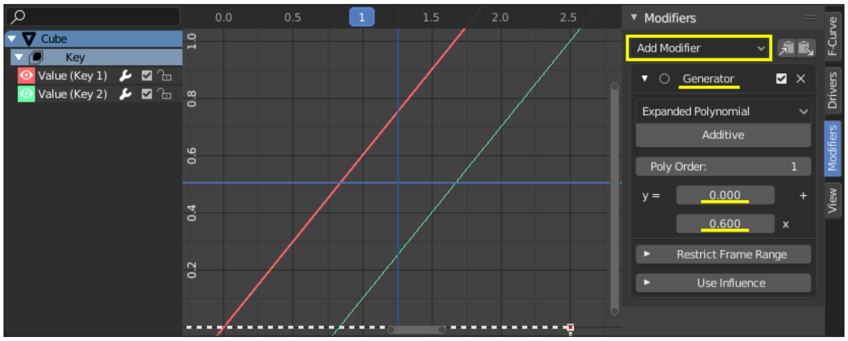
3. Configure the driver functions so that the *Value* of the shape keys (Y axis) is as desired for the bone's height (X axis).

   The driver functions should be linear, therefore, they can be defined analytically with a function of type \(y = a + bx\), where \(a\) is an offset in \(y\) and \(b\) is the slope.

   1. In the *Modifiers* tab, add a *Generator* of type *Extended Polynomial* for both drivers.

2. Play with the values of a and b so that the curves go from [0.0, 1.0] in the Y axis and from [0.0, 2.5] in the X axis. The curves should overlap in the mid area of the X axis and they should have the same slope (b).

   Possible values are *Key 1*: $y = 0.0 + 0.6x$ and *Key 2*: $y = -0.5 + 0.6x$.



Cube
Key
Value (Key 1)
Value (Key 2)
0.0
0.5
1
1.5
2.0
2.5
1.0
0.8
0.6
0.4
0.2
Modifiers
Add Modifier
Generator
Restrict Frame Range
Use Influence
F-Curve
Drivers
Modifiers
View

Expanded Polynomial

Additive

Poly Order: 1

y = 0.000 +

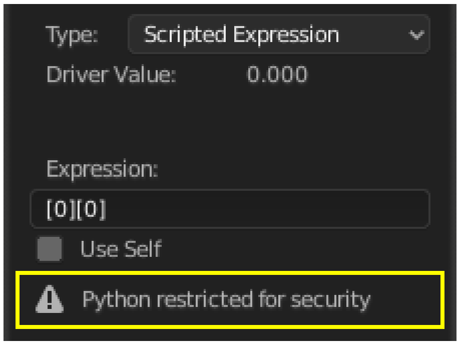
0.600 x

   Note that the functions go outside the range [0.0, 1.0] for the shape keys' *Value*, but that has no effect because *Value* is clamped in a *Range* in the *Shape Keys* panel.

Troubleshooting

Some common problems people may run into when using drivers.

Scripted Expression



Type: Scripted Expression

Driver Value: 0.000

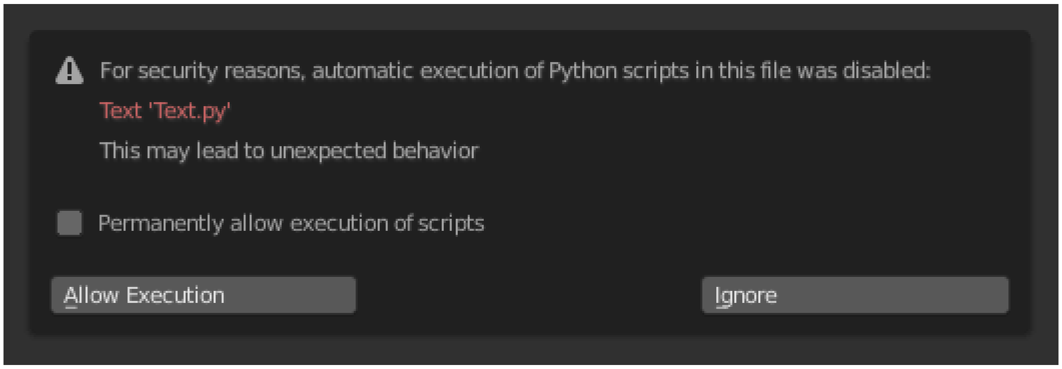
Expression:

[0][0]

☐ Use Self

Python restricted for security

A security warning in the Drivers panel.



For security reasons, automatic execution of Python scripts in this file was disabled:

Text 'Text.py'

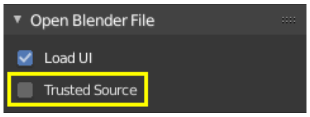
This may lead to unexpected behavior

☐ Permanently allow execution of scripts

An Auto-run warning in the Info editor's header.

By default Blender will restrict execution of Python scripts.

If using a *Scripted Expression* Driver Type that doesn't follow the Simple Expressions subset, you will have to open the file as *Trusted Source*, or set *Auto Run Python Scripts* in *Preferences* ▸ *Save & Load* ▸ *Blender Files*.

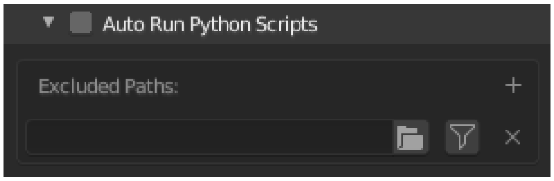


Open Blender File

☑ Load UI

☐ Trusted Source

The Trusted Source checkbox in the File Browser.



☐ Auto Run Python Scripts

Excluded Paths:

The Auto Run Python Scripts checkbox in the Preferences.

Rotational Properties are Radians

Parts of the User Interface may use different units of measurements for angles, rotation. In the Graph Editor, while working with Drivers, all angles are Radians.

Markers

Markers are used to denote frames with key points or significant events within an animation. E.g. it could be that a character's animation starts, the camera changes position, or a door opens. Markers can be given names to make them more meaningful at a quick glance. They are available in many of Blender's editors.

Note

Unlike keyframes, markers are always placed at a whole frame number, you cannot set a marker at frame 2.5.

Markers can be created and edited in the following editors:

- Graph Editor
- Dope Sheet
- NLA Editor
- Video Sequence Editor
- Timeline

Note

A marker created in one of these editors will also appear in all others that support them.

Types

Besides standard markers, *pose markers* are another type of markers, which are specific to armatures and shape keys. They are used to denote poses in the Action Editor mode and Shape Keys Editor of Dope Sheet.

Visualization

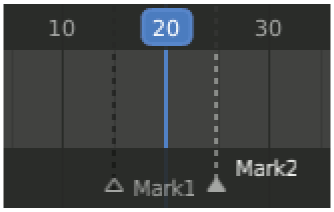
In the supported editors, if at least one is created, markers are visualized in a separate row at their bottom. This area can be disabled per editor via the *View* ▸ *Show Markers* menu option.

Note

While the markers area is disabled, markers operators are not available in that editor, and in the header the *Marker* menu is hidden.

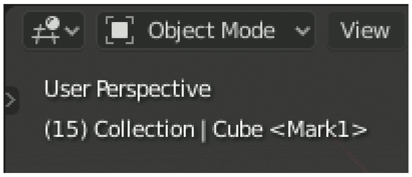
Standard

Regular markers are shown as small white triangles, empty if unselected or filled if selected, and with a dashed line that covers the editor height at the corresponding frame. If they have a name, this is shown to their right in white.



3D Viewport

The 3D Viewport does not allow you to create, edit or remove markers, but it shows their name in the Object Info in the upper left corner, when on their frame.

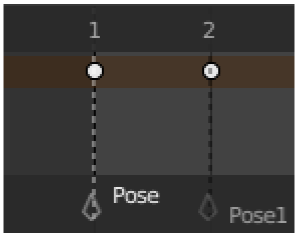


User Perspective

(15) Collection | Cube <Mark1>

Pose Markers

Pose markers show a diamond-shaped icon in the Dope Sheet. In the NLA editor pose markers are shown as a red dashed line inside the relative action strip.



Add Marker

Reference

Mode::	All modes
Menu::	*Marker* ‣ *Add Marker*
Shortcut::	`M`

The simplest way to add a marker is to move to the frame where you would like it to appear, and press `M`.

Hint

Markers can also be added during playback.

Pose Markers

If *Show Pose Markers* is checked, a pose marker and a new pose in the Old Pose Library are added.

Selecting

Reference

Mode::	All modes
Shortcut::	`LMB`

Click `LMB` on the marker's triangle to select it. Use `Shift-LMB` to select multiple markers.

In the Graph Editor, Dope Sheet, NLA Editor, Timeline, and Video Sequence Editor, you can also select all markers with `A` while hovering the mouse over the marker row, and apply selection tools on them like Box

Select, etc. (as usual, `LMB` to select, `RMB` to deselect). The corresponding options are found in the Select menu of these editors.

Editing

Duplicate Marker

Reference

Mode::	All modes
Menu::	*Marker ‣ Duplicate Marker*
Shortcut::	`Shift-D`

You can duplicate the selected markers by pressing `Shift-D`. Once duplicated, the new ones are automatically placed in select mode, so you can move them to the desired location.

Note

Note that unlike most other duplications in Blender, the names of the duplicated markers are not altered at all (no `.001` numeric counter append).

Duplicate Marker to Scene

Reference

Mode::	All modes
Menu::	*Marker ‣ Duplicate Marker to Scene…*

Duplicates the selected markers into another scene.

Delete Marker

Reference

Mode:: All modes
Menu:: *Marker ‣ Delete Marker*
Shortcut:: `X`

To delete the selected markers simply press `X`, and confirm the pop-up message with `LMB`.

Rename Marker

Reference

Mode:: All modes
Menu:: *Marker ‣ Rename Marker*
Shortcut:: `F2`

Having dozens of markers scattered throughout your scene’s time will not help you much unless you know what they stand for. You can name a marker by selecting it, pressing `F2`, typing the name, and press `Return`

Move Marker

Reference

Mode:: All modes
Menu:: *Marker ‣ Move Marker*
Shortcut:: `G`

Once you have one or more markers selected, press `G`, while hovering with the mouse over the marker bar, to move them, and confirm the move with `LMB` or `Return` (as usual, cancel the move with `RMB`, or `Esc`). Or drag them with the `LMB`.

By default, you move the markers in one-frame steps, but if you hold `Ctrl`, the markers will move in steps corresponding to 1 second (according to the scene's *FPS*).

Select

Reference

Mode:: All modes
Menu:: *Marker ‣ Select*

Convenient operators for selecting Marks; see Selecting for more information on selecting Markers.

All `A`
: Selects all Markers.

None `Ctrl-A`
: Deselects any already selected Markers.

Invert `Ctrl-I`
: Select all unselected Markers and deselects all selected Markers.

Before Current Frame
: Selects all Markers to the left of the current frame and the Marker on the current frame if it exists.

After Current Frame
: Selects all Markers to the Right of the current frame and the Marker on the current frame if it exists.

Show Pose Markers

Reference

Editor:: Dope Sheet
Mode:: Action Editor or Shape Keys Editor mode
Menu:: *Marker ‣ Show Pose Markers*

Shows markers belonging to the active action instead of scene markers.

Make Markers Local

Reference

Mode::	All modes
Menu::	*Marker ‣ Make Markers Local*

It is possible to convert standard markers into pose markers with *Marker ‣ Make Markers Local*. Note that the original marker will be gone. If you want to keep it, make a duplicate before you convert.

Jump to Next/Previous Marker

Reference

Mode::	All modes
Menu::	*Marker ‣ Jump to Next/Previous Marker*

Moves the Playhead to the next/previous marker relative to the current frame.

Bind Camera to Markers

Reference

Editor::	Timeline
Menu::	*Marker ‣ Bind Camera to Markers*
Shortcut::	`Ctrl-B`

Bind Camera to Markers is a special operator only available in the *Timeline*. The operator allows markers to be used to set the active object as the active camera.

To use this operator, select the object to become the active camera and select a marker to bind the active camera to. If no marker is selected when the operator is applied, a marker will be added. When an object is bound to a marker, the marker will be renamed to the name of the active object. These markers also have a camera icon next to the left of the name to easily distinguish them from other informative markers.

These markers can be moved to change the frame at which the active camera is changed to the object the marker is bound to.

Shape Keys

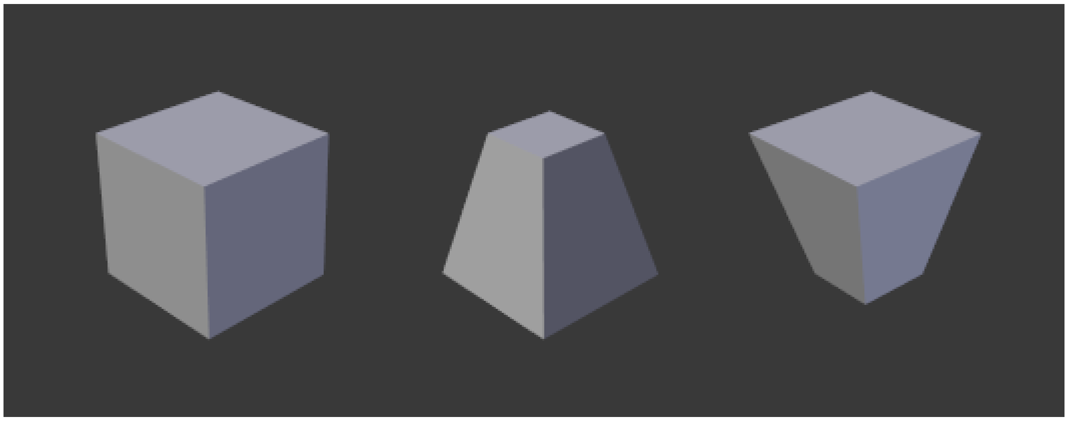
- Introduction
    - Workflow
    - Relative or Absolute Shape Keys
- Shape Keys Panel
    - Relative Shape Keys
    - Absolute Shape Keys
- Workflow
    - Relative Shape Keys
    - Absolute Shape Keys

Introduction

Shape keys are used to deform objects into new shapes for animation. In other terminology, shape keys may be called “morph targets” or “blend shapes”.

The most popular use cases for shape keys are in character facial animation and in tweaking and refining a skeletal rig. They are particularly useful for modeling organic soft parts and muscles where there is a need for more control over the resulting shape than what can be achieved with combination of rotation and scale.

Shape keys can be applied on object types with vertices like mesh, curve, surface and lattice.



Example of a mesh with different shape keys applied.

Workflow

Shape keys are authored in the Shape Keys panel which is accessed in the Object Data tab of the Properties (e.g. the Mesh tab for mesh objects).

A shape key is modified by first selecting a shape key in the panel, and then moving the object’s vertices to a new position in the 3D Viewport.

The panel has controls for affecting the current *Value* (influence, weight) of a shape. It is possible to see a shape in isolation or how it combines with others.

Adding and Removing Vertices

It is not possible to add or remove vertices in a shape key. The number of vertices and how they connect is specified by the mesh, curve, surface or lattice. A shape key merely records a position for each vertex and therefore shapes always contain all the object's vertices.

When adding a vertex, all shape keys will record it with the position in which it is created. Workflow-wise, adding and deleting vertices after creating shape keys is possible, but it is best to leave the creation of shape keys for when the mesh is finished or its topology is stable.

Adding Shape Keys

When adding a new shape key with the + button next to the list, the new shape will be a copy of the Basis shape, independently of the current result visible in the 3D Viewport.

When adding a new shape key from *Specials* ▸ *New Shape from Mix*, the shape will start of with the vertex configuration that is visible at that moment.

When doing facial animation with relative shape keys, it can be useful to first create a shape key with a complex extreme pose (e.g. anger or surprise), and then break this complex shape into components by applying a temporary vertex group to the complex shape and creating a copy with *New Shape from Mix*. This technique helps reducing conflicts between different shape keys that would otherwise produce a double effect.

Relative or Absolute Shape Keys

A mesh (curve, surface or lattice) has a stack of shape keys. The stack may be of *Relative* or *Absolute* type.

Relative

Mainly used for muscles, limb joints, and facial animation.

Each shape is defined relative to the Basis or to another specified shape key.

The resulting effect visible in the 3D Viewport, also called *Mix*, is the cumulative effect of each shape with its current value. Starting with the Basis shape, the result is obtained by **adding** each shape's weighted **relative** offset to its reference key.

Value

Represents the weight of the blend between a shape key and its reference key.

A value of 0.0 denotes 100% influence of the reference key and 1.0 of the shape key. Blender can extrapolate the blend between the two shapes above 1.0 and below 0.0.

Basis

Basis is the name given to the first (top-most) key in the stack.

The Basis shape represents the state of the object's vertices in their original position. It has no weight value and it is not keyable. This is the default *Reference Key* when creating other shapes.

Absolute

Mainly used to deform the objects into different shapes over time.

Each shape defines how the object's shape will be at *Evaluation Time* specified in its *Value*.

The resulting shape, or *Mix*, is the interpolation of the previous and next shape given the current *Evaluation Time*.

Value

Represents the *Evaluation Time* at which that shape key will be active.

Basis

Basis is the name given to the first (topmost) key in the stack.

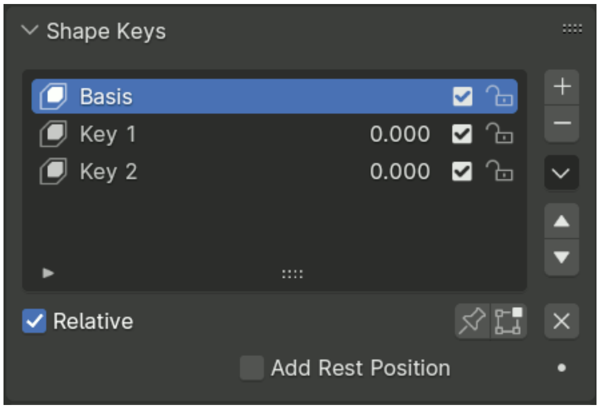
The Basis shape represents the state of the object’s vertices in their original position.

Shape Keys Panel

Reference

Editor:: Properties
Mode:: All modes
Panel:: *Object Data ‣ Shape Keys*

The Shape Keys panel is used for authoring shape keys.



Shape Keys panel.

Active Shape Key Index

A List View.

Value/Frame (number)

In Relative mode: Value is the current influence of the shape key used for blending between the shape (value=1.0) and its reference key (value=0.0). The reference key is usually the Basis shape. The weight of the blend can be extrapolated above 1.0 and below 0.0.

In Absolute mode: Value is the *Evaluation Time* at which the shape will have maximum influence.

Mute (check mark)

If unchecked, the shape key will not be taken into consideration when mixing the shape key stack into the result visible in the 3D Viewport.

Lock Shape (lock icon)

Shape keys can be locked to protect them from accidental modification due to inadvertently selecting the wrong key for editing in the list. Most common sculpt and edit mode operators and tools that move vertices abort with an error if the active shape key is locked.

> **Note**
>
> Operators that always modify all shape keys in exactly the same way, like Apply Object Transforms, don’t check shape key locks. Neither currently do most edit mode operators that modify topology, because the topology is expected to usually be finalized before shape keys are created.

Shape Key Specials

New Shape from Mix

Add a new shape key with the current deformed shape of the object. This differs from the + button of the list, as that one always copies the Basis shape independently of the current mix.

Mirror Shape Key

If your mesh is symmetrical, in *Object Mode*, you can mirror the shape keys on the X axis. This will not work unless the mesh vertices are perfectly symmetrical. Use the *Mesh* ▸ *Symmetrize* tool in *Edit Mode*.

Mirror Shape Key (Topology)

Same as *Mirror Shape Key* though it detects the mirrored vertices based on the topology of the mesh. The mesh vertices do not have to be perfectly symmetrical for this action to work.

Join as Shapes (Transfer Mix)

Transfer the current resulting shape from a different object.

Select the object to copy, then the object to copy into. Use this action and a new shape key will be added to the active object with the current mix of the first object.

Transfer Shape Key

Transfer the active shape key from a different object regardless of its current influence.

Select the object to copy, then the object to copy into. Use this action and a new shape key will be added to the active object with the active shape of the first object.

Delete All Shape Keys

Removes all Shape Keys and any effect that they had on the mesh.

Apply All Shape Keys

Saves the current visible shape to the mesh data and deletes all Shape Keys.

Relative

Set the shape keys to *Relative* or *Absolute*. See Relative or Absolute Shape Keys.

Shape Key Lock (pin icon)

Show the active shape in the 3D Viewport without blending. *Shape Key Lock* gets automatically enabled while the object is in *Edit Mode*.

Shape Key Edit Mode (edit mode icon)

If enabled, when entering *Edit Mode* the active shape key will **not** take maximum influence as is default. Instead, the current blend of shape keys will be visible and can be edited from that state.

Add Rest Position

Creates an Attribute in the vertex domain called `rest_position` which is a copy of the `position` attribute before shape keys and modifiers are evaluated. Only mesh objects support this option.

Relative Shape Keys

See Relative or Absolute Shape Keys.

With relative shape keys, the value shown for each shape in the list represents the current weight or influence of that shape in the current *Mix*.

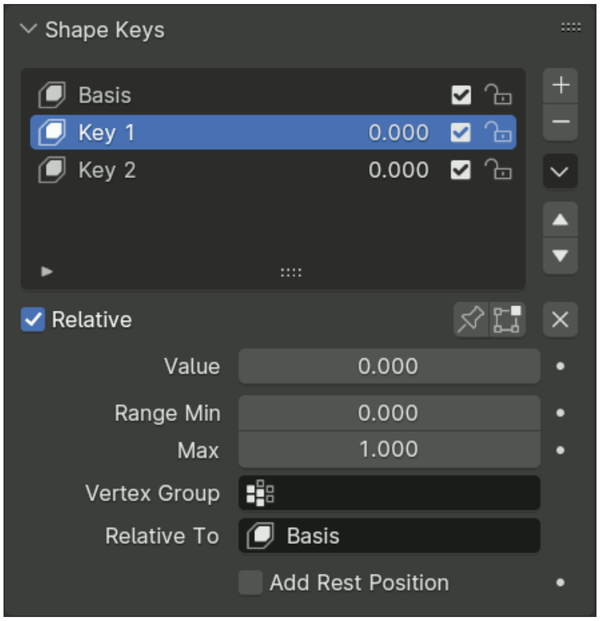
(Clear Shape Keys)

Set all influence values, or weights, to zero. Useful to quickly guarantee that the result shown in the 3D Viewport is not affected by shapes.

Value

The weight of the blend between the shape key and its reference key (usually the Basis shape).

A value of 0.0 denotes 100% influence of the reference key and 1.0 of the shape key.



Basis ☑
Key 1 0.000 ☑
Key 2 0.000 ☑

☑ Relative

Value 0.000
Range Min 0.000
Max 1.000
Vertex Group
Relative To Basis
☐ Add Rest Position

Relative Shape Keys options.

Range

Minimum and maximum range for the influence value of the active shape key. Blender can extrapolate results when the *Value* goes lower than 0.0 or above 1.0.

Vertex Group

Limit the active shape key deformation to a vertex group. Useful to break down a complex shape into components by assigning temporary vertex groups to the complex shape and copying the result into new simpler shapes.

Relative To

Select the shape key to deform from. This is called the *Reference Key* for that shape.

Note

Rather than storing offsets directly, internally relative keys are stored as snapshots of the mesh shape. The relative deformation offsets are computed by subtracting *Reference Key* from that snapshot.

Therefore, replacing the *Reference Key* has the effect of subtracting the difference between the new and old reference from the relative deformation of the current key.

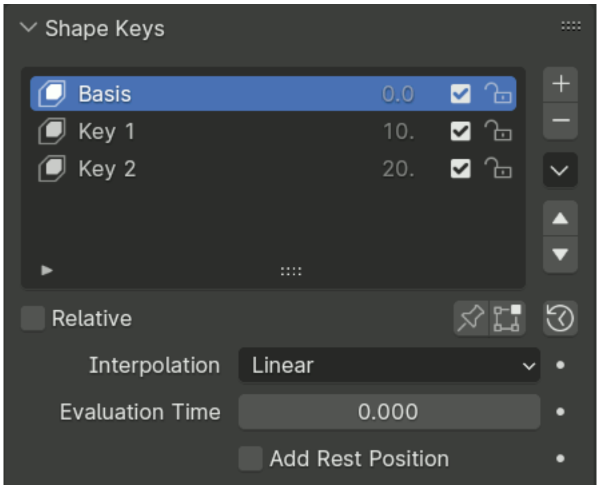
Absolute Shape Keys

See Relative or Absolute Shape Keys.

With absolute shape keys, the value shown for each shape in the list represents the *Evaluation Time* at which that shape key will be active.

Re-Time Shape Keys (clock icon)

Absolute shape keys are timed, by order in the list, at a constant interval. This button resets the timing for the keys. Useful if keys were removed or re-ordered.



Shape Keys

Basis	0.0	☑
Key 1	10.	☑
Key 2	20.	☑

☐ Relative

Interpolation: Linear

Evaluation Time: 0.000

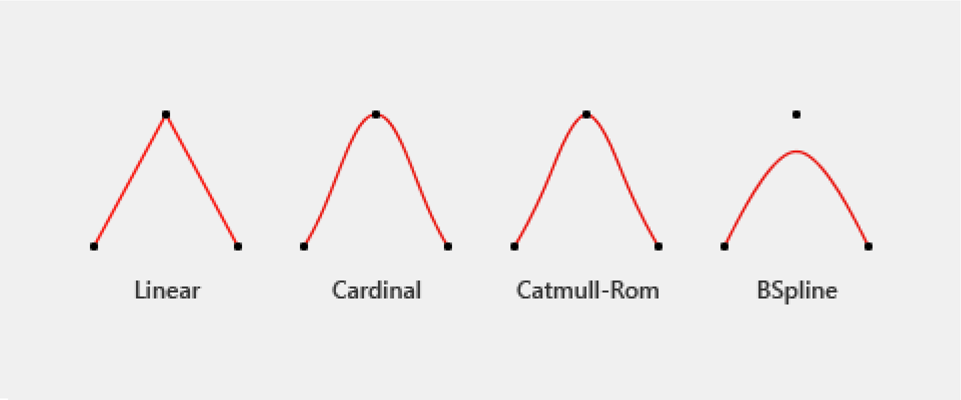
☐ Add Rest Position

Absolute Shape Keys options.

Interpolation

Controls the interpolation between shape keys.

Linear, Cardinal, Catmull-Rom, B-Spline



Different types of interpolation.

The red line represents interpolated values between keys (black dots).

Evaluation Time

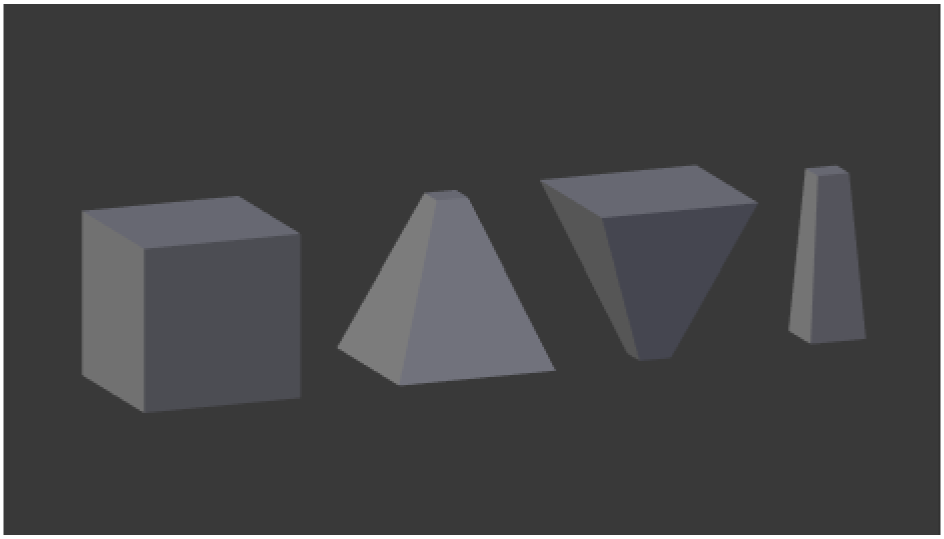
Controls the shape key influence. Scrub to see the effect of the current configuration. Typically, this property is keyed for animation or rigged with a driver.

Workflow

Relative Shape Keys

1. In *Object Mode*, add a new shape key via the *Shape Key* panel with the + button.
2. “Basis” is the rest shape. “Key 1”, “Key 2”, etc. will be the new shapes.
3. Switch to *Edit Mode*, select “Key 1” in the *Shape Key* panel.
4. Deform mesh as you want (do not remove or add vertices).
5. Select “Key 2”, the mesh will be changed to the rest shape.
6. Transform “Key 2” and keep going for other shape keys.
7. Switch back to *Object Mode*.
8. Set the *Value* for “Key 1”, “Key 2”, etc. to see the transformation between the shape keys.

In the figure below, from left to right shows: “Basis”, “Key 1”, “Key 2” and mix (“Key 1” `1.0` and “Key 2” `0.8`) shape keys in Object Mode.

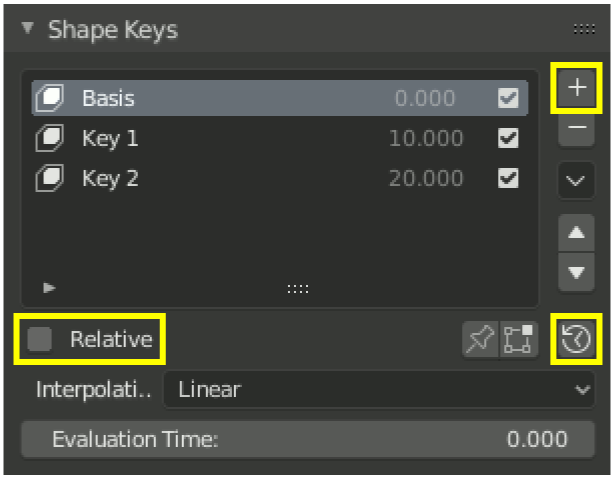


Relative shape keys example.

For more practical examples, see how to combine shape keys and drivers.

Absolute Shape Keys

1. Add sequence of shape keys as described above for relative shape keys.
2. Uncheck the *Relative* checkbox.
3. Click the *Reset Timing* button.
4. Switch to *Object Mode*.
5. Drag *Evaluation Time* to see how the shapes succeed one to the next.



Basis	0.000	☑
Key 1	10.000	☑
Key 2	20.000	☑

☐ Relative

Interpolati.. Linear

Evaluation Time: 0.000

Absolute shape keys workflow.

By adding a driver or setting keyframes to *Evaluation Time* you can create an animation.

See also

Shape Key Operators

There are two modeling tools used to control shape keys and are found in Edit Mode.

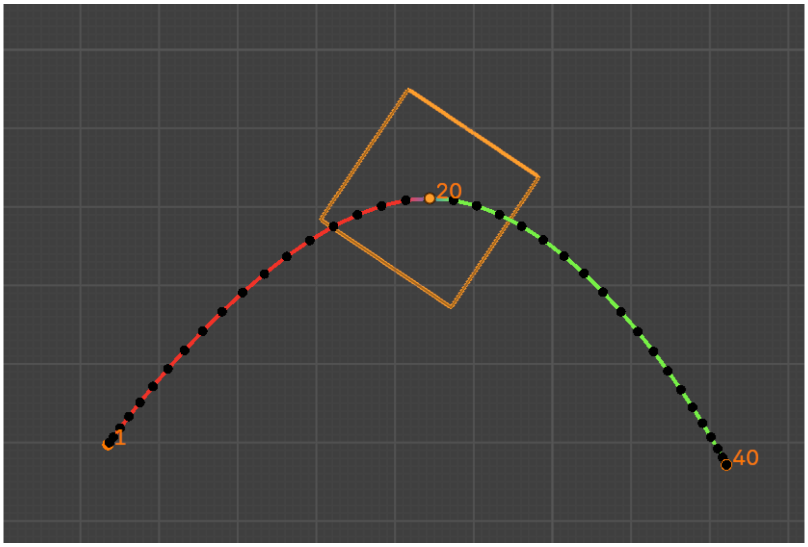
Motion Paths

Reference

Editor::	3D Viewport, Properties
Mode::	Object Mode
Panel::	*Properties ‣ Object Properties ‣ Motion Paths*

Reference

Editor::	3D Viewport, Properties
Mode::	Pose Mode
Panel::	*Properties ‣ Armature ‣ Motion Paths*
Menu::	*Pose ‣ Motion Paths*



An animated cube with its motion path displayed.

The Motion Paths tool allows you to visualize the motion of points as paths over a series of frames. These points can be object origins and bone joints.

To create or remove motion paths, it is necessary to first select the bones. Then:

1. To show the paths (or update them, if needed), click on the *Calculate Path* button.
2. To hide the paths, click on the *Clear Paths* button.

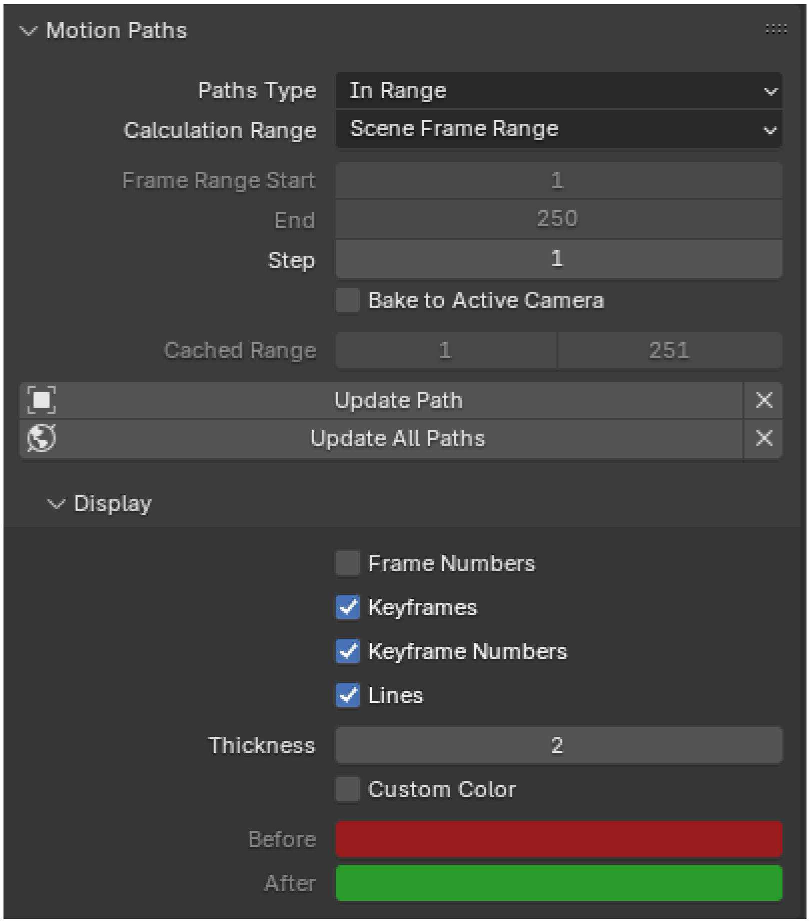
Note

Remember that only selected bones and their paths are affected by these actions!

The paths are shown in red for the section in the past and green for the section in the future. These colors follow the user preference options “Before Current Frame” and “After Current Frame”, which can be found in the 3D Viewport section. Each frame is displayed by a small dot on the paths.

The paths are automatically updated when you edit your poses/keyframes, and they are also active during animation playback. Playing the animation affects the paths only when using the *Around Frame* type.

Options



Paths Type: In Range
Calculation Range: Scene Frame Range
Frame Range Start: 1
End: 250
Step: 1
☐ Bake to Active Camera
Cached Range: 1 251

☐ Frame Numbers
☑ Keyframes
☑ Keyframe Numbers
☑ Lines
Thickness: 2
☐ Custom Color
Before: 🟥
After: 🟩

The Motion Paths panel in the Armature tab.

Paths Type

Type of range to show for Motion Paths.

Around Frame::	Display paths of points within a fixed number of frames around the current frame. When you enable this button, you get paths for a given number of frames before and after the current one
In Range::	Display paths of points within specified range.

Calculation Range

The range of the motion path. Only active when *Paths Type* is set to *In Range*. Changing this option only takes effect when updating the path, via the *Update Path* or *Update All Paths* buttons.

All Keys::	Generate a motion path ranging from the first keyframe to the last. Only the keys of the active object / bone are considered.
Selected Keys::	Same as *All Keys* except that it ranges from the first to the last *selected* keyframe.
Scene Frame Range::	Use the start & end frames of the scene, or the preview range if active.
Manual Range::	Manually set the start and end frame.

Frame Range Start, End

Starting and Ending frame of range of paths to display/calculate (not for the *Around Frame* type).

Although the start and end frame are always editable, updating the motion path will change these according to the *Calculation Range* setting. To ensure your chosen frame range is maintained, choose *Manual Range* there.

Frame Range Before, After

Number of frames to show before and after the current frame (only for the *Around Frame* type).

Step

Allows displaying one point for every *n* frames on the path. Mostly useful when you enable the frame number display (see below), to avoid cluttering the 3D Viewport.

Bake to Active Camera

When enabled the motion path is calculated in screen space for the active scene camera. Note that the resulting motion path will only be useful for that single camera. Switching cameras using markers is not supported. It will only bake to the camera that is active when the bake was started.

Cache/Bone Cache From, To

These are the start/end frames of the range in which motion paths are shown. You cannot modify this range without deleting the motion path